SCICHART v3.0 (BETA) Release Note

Thank you for downloading SciChart v3.0! A lot has changed since v2.x so please see the New Features and Improvements section and Update Instructions for users of v2.x at the end of this document if you are an existing user of SciChart.

Contents

New Features and Improvements .................................................. Page 1
Update Instructions from v2.x ......................................................... Page 8
License Key Upgrades ................................................................. Page 9
A note about support.scichart.com .............................................. Page 10

New Features & Improvements

1. ResamplingMode.MinMaxUneven

- Allows resampling of unevenly spaced data
- Previously SciChart required that data was evenly spaced in order to resample using ResamplingMode.MinMax and maintain accurate visual output.
- No more rendering errors when Resampling is used with unevenly spaced data

Usage:

<SciChart:FastLineRenderableSeries ResamplingMode="MinMaxWithUnevenSpacing"/>

2. ResamplingMode.Auto (Default)

- Auto-detection of data distribution (sorted, unsorted, evenly spaced, unevenly spaced)
- Best selection of fastest/most accurate resampling mode for your data.
- No more rendering errors due to MinMax resampling and data-distribution

Usage:

<!-- Note that ResamplingMode.Auto is the default setting -->

3. UnsortedXyDataSeries is no longer needed

- XyDataSeries now detects whether data is sorted or unsorted.
- No need to specify if data is unsorted, however performance is still better if it is.
- HitTest and resampling algorithms are changed accordingly.
4. MultiTouch support

- Zooming, panning, axis manipulations, cursors, annotation manipulations via touch screens in WPF and SL

Usage:

Multitouch is enabled by default in SciChart and all existing modifiers now have touch-support. To get a touch-specific pinch zoom action please use the PinchZoomModifier.

```xml
<!-- Adding the PinchZoomModifier gives SciChart the ability to zoom by pinching on a multi-touch screen -->
<sciChart:SciChartSurface.ChartModifier>
    <sciChart:ModifierGroup>
        <sciChart:PinchZoomModifier />
        <sciChart:ZoomPanModifier />
        <sciChart:ZoomExtentsModifier />
    </sciChart:ModifierGroup>
</sciChart:SciChartSurface.ChartModifier>
```

5. Axis Styling improvements

- Chart Titles, Axis Titles and Labels can now be styled
- Can set font size, font style, font weight on Axis labels
- Now supports optional rotation of axis tick labels (e.g. 90 degrees)

Usage:

```xml
<s:DateTimeAxis>
    <s:DateTimeAxis.TickLabelStyle>
        <Style TargetType="s:DefaultTickLabel">
            <Setter Property="Foreground" Value="Blue" />
            <Setter Property="FontSize" Value="25" />
            <Setter Property="LayoutTransform">
                <Setter.Value>
                    <RotateTransform Angle="45" />
                </Setter.Value>
            </Setter>
        </Style>
    </s:DateTimeAxis.TickLabelStyle>
</s:DateTimeAxis>
```

Or

```xml
<s:NumericAxis>
    <s:NumericAxis.TickLabelStyle>
        <Style TargetType="s:NumericTickLabel">
            <Setter Property="Foreground" Value="Blue" />
            <Setter Property="FontSize" Value="25" />
        </Style>
    </s:NumericAxis.TickLabelStyle>
</s:NumericAxis>
```
6. Axis Tick Algorithm Improvements

- AxisBase.TickProvider API to override in-built axis tick generation algorithms
- Improved axis culling / adjacency algorithm

Usage:

```csharp
// Tickprovider Declaration
public class CustomTickProvider : TickProvider<DateTime>
{
    public override DateTime[] GetMinorTicks(IAxisParams axis)
    {
        return new DateTime[]{ new DateTime(2014, 1, 5), new DateTime(2014, 1, 15), new DateTime(2014, 1, 25) };
    }

    public override DateTime[] GetMajorTicks(IAxisParams axis)
    {
        return new DateTime[]{ new DateTime(2014, 1, 1), new DateTime(2014, 1, 10), new DateTime(2014, 1, 20) };
    }
}

// Using a TickProvider in code
var axis = new DateTimeAxis();
axis.TickProvider = new CustomTickProvider();

OR

<!-- Where CustomTickProvider is declared as a Resource -->
<sciChart:DateTimeAxis TickProvider="{StaticResource CustomTickProvider}"/>
```

7. Static Axis

- AxisBase.IsStaticAxis fixes all tick labels at static positions, and updates the text-values instead
- Perfect for static charts, or real-time charts where fixed-position labels are needed

Usage:

```xml
<!-- Create Y Axis with IsStaticAxis flag -->
<sciChart:SciChartSurface.YAxes>
    <sciChart:NumericAxis IsStaticAxis="True"/>
</sciChart:SciChartSurface.YAxes>
```
8. Axis VisibleRangeLimit

- Does not allow the axis to expand beyond the given AxisBase.VisibleRangeLimit.
- Provides a reset-point for double-click Zoom Extents, overriding the built-in auto-ranging mechanism

Usage:

```xml
<!-- Bind X and Y Axis VisibleRangeLimit to ViewModel properties -->
<s:SciChartSurface.XAxis>
  <s:NumericAxis VisibleRangeLimit="{Binding XLimit}"/>
</s:SciChartSurface.XAxis>

<s:SciChartSurface.YAxis>
  <s:NumericAxis VisibleRangeLimit="{Binding YLimit}"/>
</s:SciChartSurface.YAxis>

// ViewModel properties declared as
private DoubleRange _xLimit;

public DoubleRange XLimit
{
    get { return _xLimit; }
    set
    {
        if (_xLimit != value) return;
        _xLimit = value;
        OnPropertyChanged("XLimit");
    }
}

private DoubleRange _yLimit;

public DoubleRange YLimit
{
    get { return _yLimit; }
    set
    {
        if (_yLimit != value) return;
        _yLimit = value;
        OnPropertyChanged("YLimit");
    }
}
```
9. Xml Serialization Support

- Serialize Chart Settings to/from XML
- Serialize Annotation Positions to/from XML

Usage:

```csharp
// Use this code to Serialize just annotations
private string SerializeAnnotations(AnnotationCollection annotationCollection)
{
    var stream = new MemoryStream();
    var serializer = new XmlSerializer(typeof(AnnotationCollection));
    serializer.Serialize(stream, annotationCollection);
    stream.Position = 0;
    string xmlText = new StreamReader(stream).ReadToEnd();
    return xmlText;
}

// Use this code to Deserialize just annotations
private AnnotationCollection DeserializeAnnotations(string xmlText)
{
    var stream = new MemoryStream(Encoding.ASCII.GetBytes(xmlText));
    var serializer = new XmlSerializer(typeof(AnnotationCollection));
    stream.Position = 0;
    var annotationCollection = (AnnotationCollection)serializer.Deserialize(stream);
    return annotationCollection;
}

// Use this code to Serialize/Deserialize a SciChartSurface
private void SerializeChart()
{
    var beforeChart = new SciChartSurface();
    // ...
    var stream = new MemoryStream();
    var serializer = new XmlSerializer(typeof(SciChartSurface));
    serializer.Serialize(stream, beforeChart);
    var afterChart = (SciChartSurface)serializer.Deserialize(stream);
}
```

10. SciChartSurface.MaxFrameRate.

- Limit frame-rate to a maximum Hertz, or leave null for default (unlimited).
- Useful for strip charts (e.g. CPU, Temperature monitoring) in process control
- Note low values, e.g. 1, will not only limit redrawing but frame-rate when responding to mouse/zoom events

Usage:

```csharp
sciChartSurface.MaxFrameRate = 24;
```

```xml
<SciChart:SciChartSurface MaxFrameRate="24"/>
```
11. Save Chart to Bitmap

- Export to BitmapSource, File built in to SciChartSurface.
- Supports file export to PNG, JPEG, BMP

Usage:

```csharp
var surface = new SciChartSurface();
//...

// Export to in-memory bitmap
var bitmapSource = surface.ExportToBitmapSource();

// Export directly to file
surface.ExportToFile("Filename.png", ExportType.Png);
```

12. Heatmap Improvements

- Implemented HitTest for Heatmap, allowing RolloverModifier, Cursors.
- GradientBrush support for Heatmap colour maps
- New improved HeatmapColourMap control

Usage:

```csharp
<s:SciChartSurface.RenderableSeries>
  <s:FastHeatMapRenderableSeries x:Name="heatmapSeries"
    Opacity="0.5"
    Maximum="200">
    <!-- Defining the Color Map using WPF LinearGradientBrush -->
    <s:FastHeatMapRenderableSeries.ColorMap>
      <LinearGradientBrush>
        <GradientStop Offset="0" Color="DarkBlue" />
        <GradientStop Offset="0.2" Color="CornflowerBlue" />
        <GradientStop Offset="0.4" Color="DarkGreen" />
        <GradientStop Offset="0.6" Color="Chartreuse" />
        <GradientStop Offset="0.8" Color="Yellow" />
        <GradientStop Offset="1" Color="Red" />
      </LinearGradientBrush>
    </s:FastHeatMapRenderableSeries.ColorMap>
    <!-- Defining the RolloverMarkerTemplate for use with Rollover -->
    <s:FastHeatMapRenderableSeries.RolloverMarkerTemplate>
      <ControlTemplate>
        <s:TemplatableObject.ContentTemplate>
          <DataTemplate>
            <Ellipse Width="9" Height="9"
              Fill="Red"
              Stroke="Blue"
              StrokeThickness="2" />
          </DataTemplate>
        </s:TemplatableObject.ContentTemplate>
      </ControlTemplate>
    </s:FastHeatMapRenderableSeries.RolloverMarkerTemplate>
  </s:FastHeatMapRenderableSeries>
</s:SciChartSurface.RenderableSeries>
```
13. API Improvements to ViewportManager

- New API Functions added to ViewportManager to directly control the SciChartSurface.
- ViewportManagerBase implements ISciChartController, defined as

```csharp
public interface ISciChartController : ISuspendable, IInvalidatableElement {
    /// <summary>
    /// An interface to a subset of methods on the SciChartSurface.
    /// </summary>
    public interface ISciChartController :

    /// <summary>
    /// Zooms the chart to the extents of the data, plus any X or Y Grow By fraction set on the X and Y Axes
    /// </summary>
    void ZoomExtents();

    /// <summary>
    /// Zooms to extents with the specified animation duration
    /// </summary>
    /// <param name="duration">The duration of animation when zooming to extents</param>
    void AnimateZoomExtents(TimeSpan duration);

    /// <summary>
    /// Zooms the chart to the extents of the data in the Y-Direction, accounting for the current data in view in the X-direction
    /// </summary>
    void ZoomExtentsY();

    /// <summary>
    /// Zooms the chart to the extents of the data in the Y-Direction, accounting for the current data in view in the X-direction
    /// </summary>
    void AnimateZoomExtentsY(TimeSpan duration);

    /// <summary>
    /// Zooms the chart to the extents of the data in the X-Direction
    /// </summary>
    void ZoomExtentsX();

    /// <summary>
    /// Zooms the chart to the extents of the data in the X-Direction
    /// </summary>
    void AnimateZoomExtentsX(TimeSpan duration);
}
```

Usage:

- Declare an instance of DefaultViewportManager in your ViewModel.
- Bind SciChartSurface.ViewportManager to this.
- Now you can control the SciChartSurface from the ViewModel via the ViewportManager methods above.
Performance Tuning

With great power comes great responsibility! Every release of SciChart has more features, more bugs fixed, and therefore more potential to be slower than the last. In SciChart v3.0 we have made slight but important improvements to performance by:

- Reducing the load of the Garbage Collector by reducing boxing/unboxing deep inside the renderer core.
- Reducing the load of the CPU by reducing redundant NaN / null checks deep in the renderer core.

We aim to keep working on this area and like a race car, shave off the extra milliseconds wherever we find them!

Update Instructions for users of v2.x

Several API changes have been made. Please follow these upgrade instructions if you experience one of the following compilation or runtime errors:

The following code will no longer compile:

Changes to Heatmap API

```
<scichart:FastHeatMapRenderableSeries x:Name="heatmapSeries"
    MaxColor="#FF22AA"
    MiddleColor="CornflowerBlue"
    MinColor="DarkBlue"
    Opacity="0.5"
    ScaleFactor="0.005" />
```

MaxColor, MiddleColor, MinColor, ScaleFactor have been replaced with the ColorMap feature described above.

Changes to ILabelFormatter interface

ILabelFormatter has two new methods on it. If you are using custom LabelFormatters in your code you will need to implement these methods, or, you will need to derive from LabelFormatterBase, which provides default implementations below.

```csharp
/// <summary>
/// Creates a <see cref="ITickLabelViewModel"/> instance, based on the data-value passed in.
/// Invokes <see cref="FormatLabel"/> to format the specified data-value passed in.
/// </summary>
public virtual ITickLabelViewModel CreateDataContext(IComparable dataValue) {
    return UpdateDataContext(new DefaultTickLabelViewModel(), dataValue);
}
```
/// <summary>
/// Updates existing <see cref="ITickLabelViewModel" />, based on the data-value passed in.
/// Invokes <see cref="FormatLabel"/> to format the specified data-value passed in.
/// </summary>
/// <param name="labelDataContext">The instance to update</param>
/// <param name="dataValue">The data-value to format</param>
public virtual ITickLabelViewModel UpdateDataContext(ITickLabelViewModel labelDataContext, IComparable dataValue)
{
    var formatted = FormatLabel(dataValue);
    labelDataContext.Text = formatted;

    return labelDataContext;
}

UnsortedXyDataSeries is obsolete

If you receive a compilation error on using UnsortedXyDataSeries, simply replace it for XyDataSeries. It should now ‘just work’.

License Key Upgrades

SciChart v3.0 ships with a 30-day trial key embedded in the application. If you’re an existing customer of SciChart with an active support subscription, you will automatically qualify for an upgrade to SciChart v3.0 and your license key should just work.

If your support has expired and you would like to renew your support subscription, please contact sales@scichart.com and we will be glad to assist.
A Note About support.scichart.com

As well as being busy creating functional requirements for SciChart, we have been building a lot of non-functional elements of a software business. One of these is the new support site, which will go live very soon at http://support.scichart.com.

Soon all email sent to sales@scichart.com and support@scichart.com will be routed through the support site, creating tickets for each email. We will no longer be handling email directly as a support channel. Too many requests get lost.

We hope that the new support site enables us to deliver high-quality support as well as taking our reliability and response times to new levels!