BIRT Chart Types Enhancements

Functional Specifications

Draft 3: August 12, 2005

Abstract

This document describes the chart types support for BIRT 2.0. That includes new chart types and improvement of existing chart types

Document Revisions

Draft	Date	Primary Author(s)	Description of Changes
1	July 13, 2005	David Michonneau	Initial Draft
2	August 5, 2005	David Michonneau	Extended 3D rotation to three axes. Changed anchor to position for fitting curve label. FittingCurve now applies to Series classes. Added Line Chart coloring from Palette. Added Mockups
3	August 12, 2005	David Michonneau Sheldon Lee-Loy	Modified FittingCurve's Label Position back to Anchor. Merged Sheldon's Dial Spec into latest spec. Updated Builder Mockups and added some new ones for Dial type.

Table of Contents

1. Introduction

Chart types are a combination of series types and chart attributes. They can be divided into two categories: charts with axis and chart without axis (such as pie charts). New series types will be provided, as well as improvements of the existing ones. Some new chart attributes will also be added to support the new types.

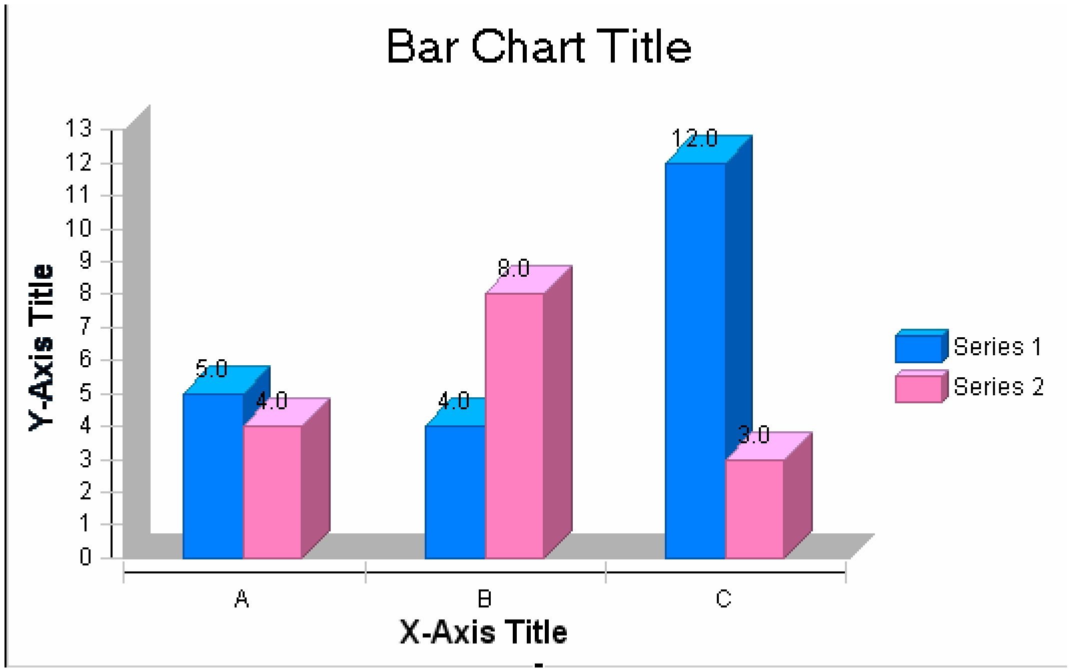
This document will focus on the functional specifications of the new and improved types and also list the model/API changes.

2. 3D Charts

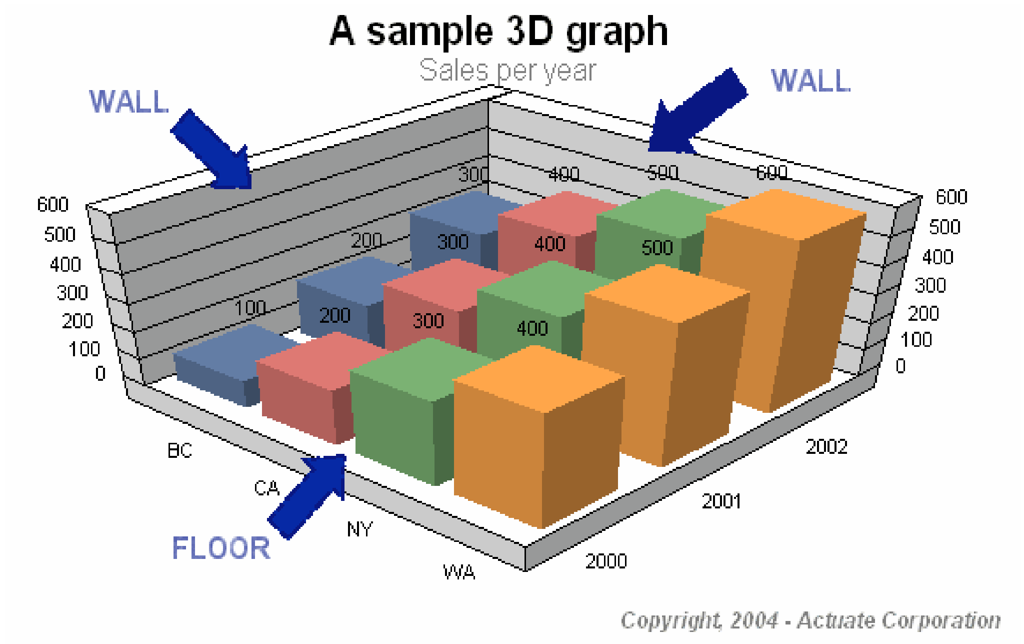
2.1 3D Definition

BIRT 1.0.1 support Charts in 2D and 2D with depth. 2D with depth (sometimes mentioned as 2.5D) is not to be confused with real 3D Charts. Here is a diagram outlining the difference between the two:

Chart 2D with depth



3D Chart



The 3D chart not only displays the series with depth: each serie is placed on a different plan. Additionally it is also possible to rotate the 3D chart from any angle. Walls and floors are also displayed to enhance the 3D appearance, but they can be hidden.

2.2 Chart types in 3D

Here is the list of chart types where you can apply 3D:

- Bar Charts
- Line Charts
- Area Charts

2.3 API

2.3.1 3D attribute

Although the current engine cannot render 3D charts yet, the existing model API support the 3D attribute. The Chart interface has two methods to set/get the chart dimension (now used for 2D/2D with depth):

```
public interface Chart extends Eobject
{
…
ChartDimension getDimension();
void setDimension(ChartDimension value);
…
}
```

The 3D value is represented by ChartDimension.THREE_DIMENSIONAL_LITERAL

2.3.2 Angle attribute

The angle attribute to rotate the chart will be added in ChartWithAxes interface through 2 methods:

```
public interface ChartWithAxes extends Chart
{
…
   3DAngles get3DRotation();
   void set3DRotation( 3DAngles angle);
…
}
public interface 3DAngles
{
   double getXAngle();
   double getYAngle();
   double getZAngle();
   void setXAngle( double angleInDegrees );
   void setYAngle( double angleInDegrees );
   void setZAngle( double angleInDegrees );
}
```

Each of the rotation angle will be applied clockwise, and must be within the 0-90 range. The 3DAngles property is only effective for 3D charts and will be ignored otherwise.

2.3.3 Floor/Wall attributes

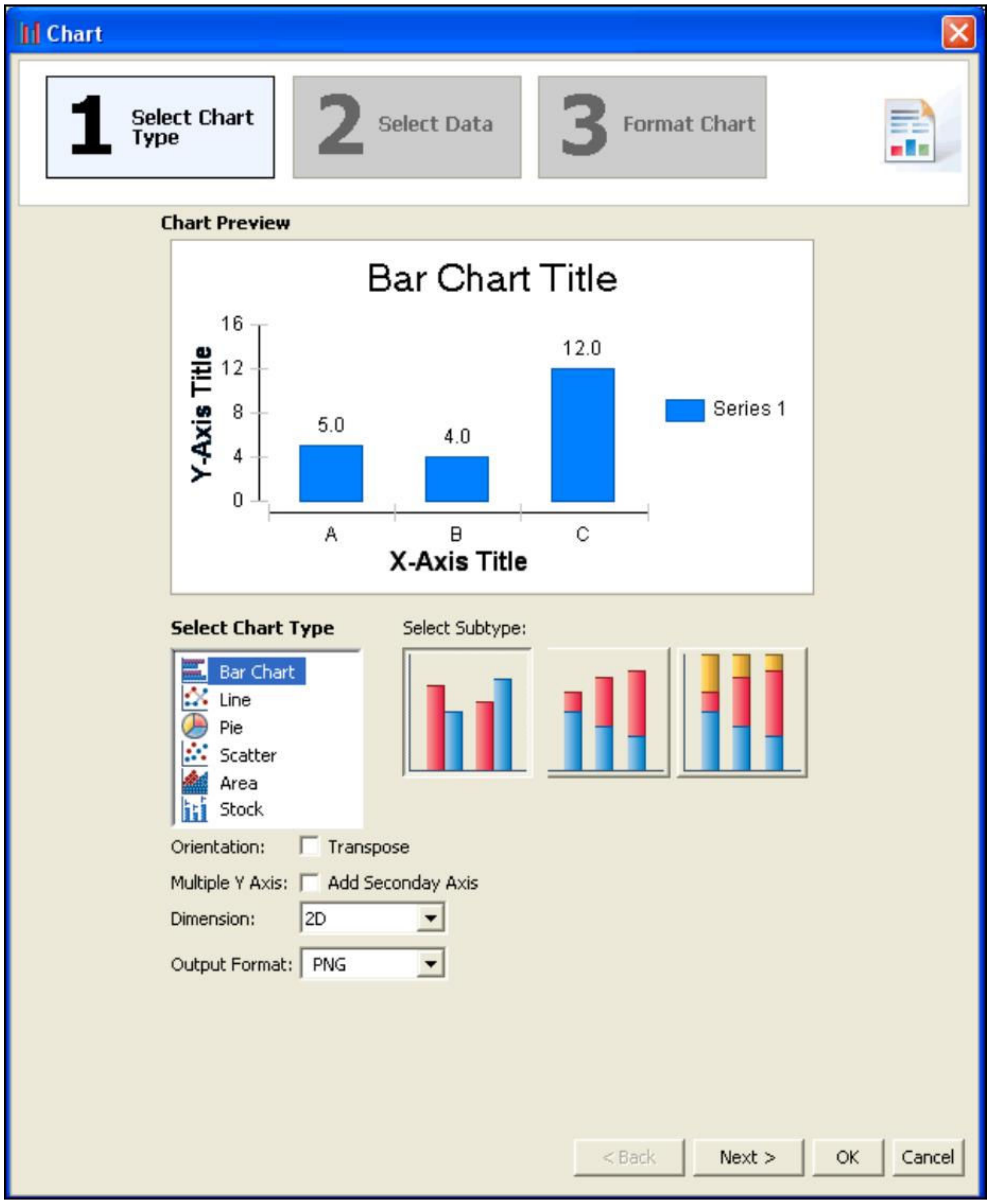
The floor and wall attributes are already defined in ChartWithAxes interfaces:

```
   public interface ChartWithAxes extends Chart
   {
…

   Fill getFloorFill();
   void setFloorFill(Fill value);
   …
   Fill getWallFill();
   void setWallFill(Fill value);
   …
   }
```

2.4 Chart Builder

The 3D attribute will be added to the Dimension combo list in the Chart Builder, which now holds the 2D and 2D with depth values. It will only be available when bar/line/area chart type is selected.



Orientation: ☐ Transpose

Multiple Y Axis: ☐ Add Seconday Axis

Dimension: 2D

Output Format: PNG

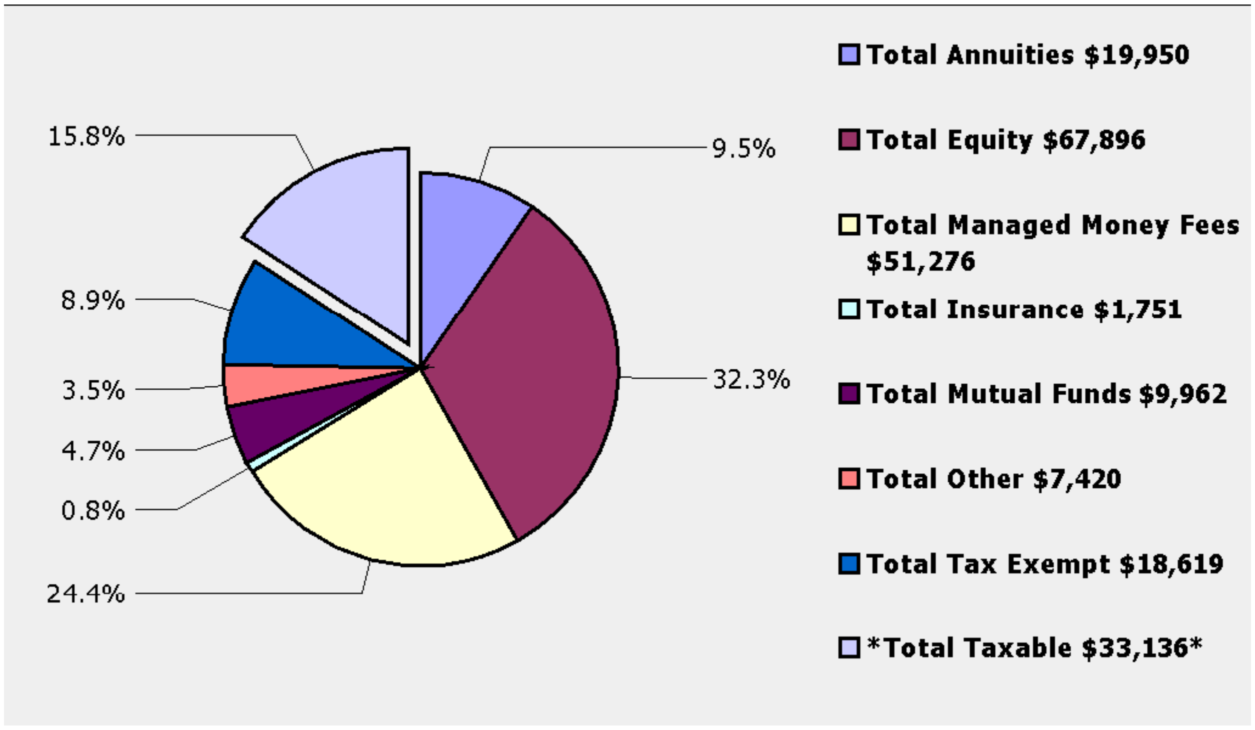
3. Pie Chart

Two improvements will be added to the Pie Chart type: explosion based on expression, and a minimum slice feature.

3.1 Explosion

Pie Charts can be exploded through an explosion attribute, but this applies uniformly to all the slices. Here is how it looks like:

The new feature will allow setting the explosion of the slices based on an expression: this way the user can highlight some particular slices. Here is an example of a chart with only one exploded slice:



The expression can access, among others, the value of the current slice (absolute or in percentage), as well as the value of the base serie it represents. TBD

If no expression is set, the explosion will apply to all slices. If the expression is set, the explosion will only apply to slices where the expression evaluates to true.

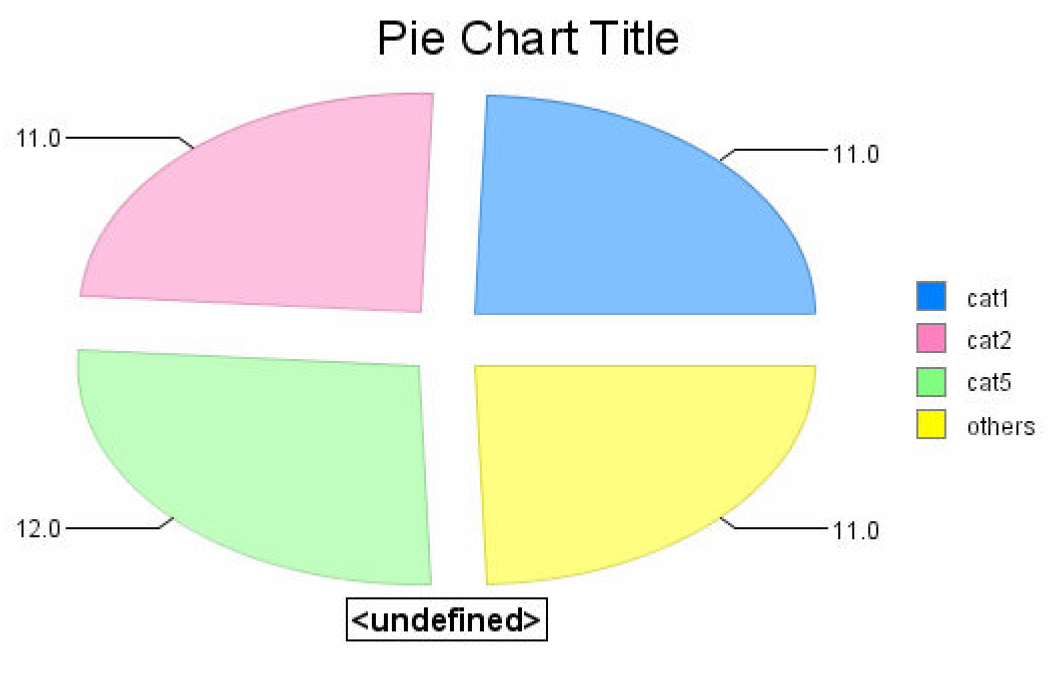
3.1.1 API

A new method setExplosionExpression will be added to PieSeries interface:

```
public interface PieSeries extends Series
{
…
   public void setExplosionExpression( String expression );
   public String getExplosionExpression( );
…
}
```

3.2 Minimum Slice

With this feature, the user has the ability to specify the minimum value (absolute or in percentage) for a slice. Values under that minimum will be aggregated under an “others” label. In this example, the min value is 10, “others” is the sum of 6 (cat3) and 4 (cat4):



3.2.1 API

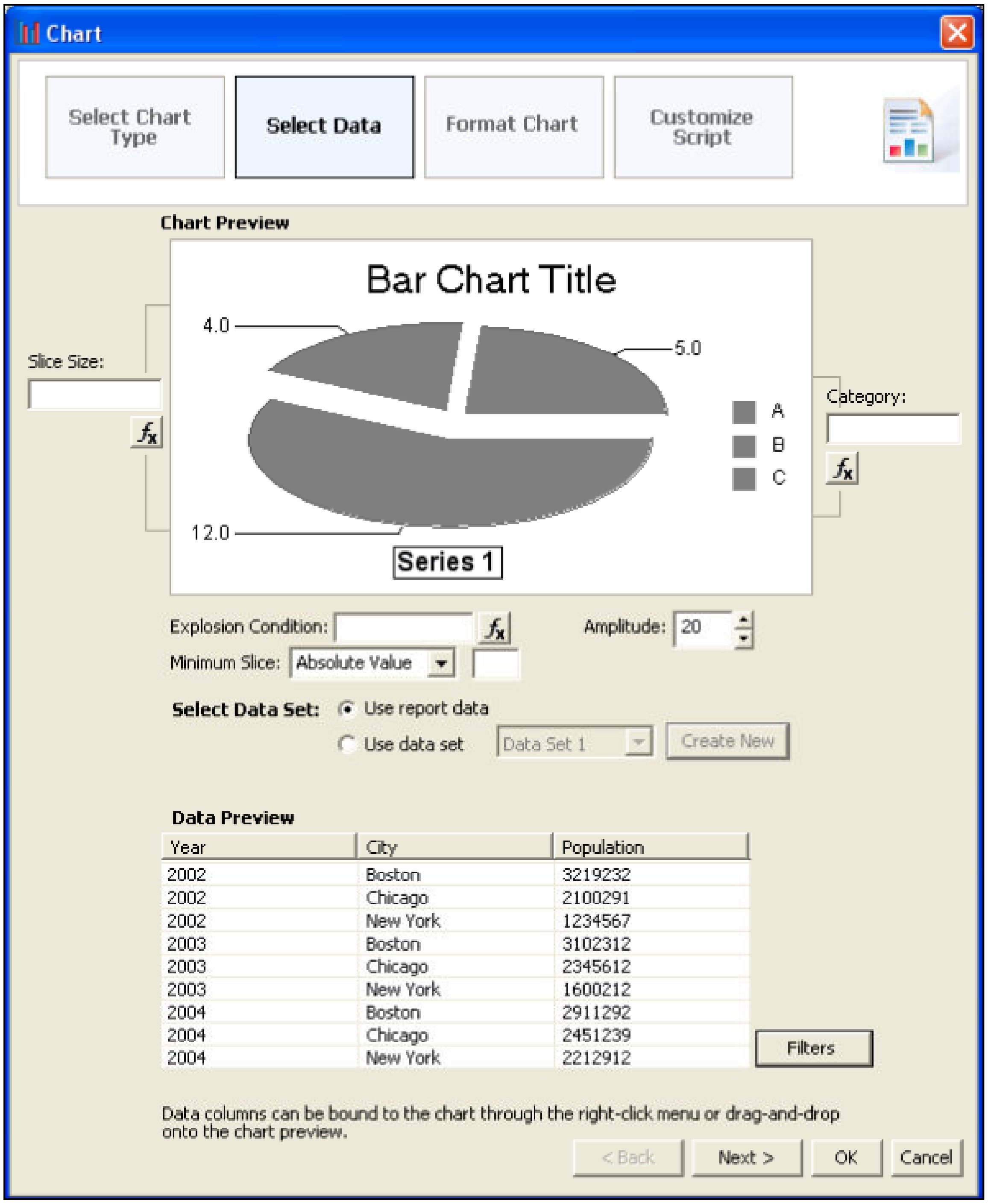
Several methods will added to the ChartWithoutAxes interfaces in order to set/get:

- the value of the minimum slice
- the value type of the minimum slice (percent vs absolute). Default is absolute
- the label of the minimum slice. Default is “others”, internationalized

This feature will be applicable to all charts without axes.

```
public interface ChartWithoutAxes extends Chart
{…
   public void setMinSlice( double value );
   public double getMinSlice( double value );
   public void setMinSlicePercent( boolean isPercent );
   public boolean getMinSlicePercent( );
   public void setMinSliceLabel( String label );
   public String getMinSliceLabel( );
…
}
```

3.3 Pie Chart Builder



Explosion Condition:

Amplitude: 20

Minimum Slice: Absolute Value

Select Data Set: (● Use report data) ○ Use data set Data Set 1

Data Preview

Year	City	Population
2002	Boston	3219232
2002	Chicago	2100291
2002	New York	1234567
2003	Boston	3102312
2003	Chicago	2345612
2003	New York	1600212
2004	Boston	2911292
2004	Chicago	2451239
2004	New York	2212912

Data columns can be bound to the chart through the right-click menu or drag-and-drop onto the chart preview.

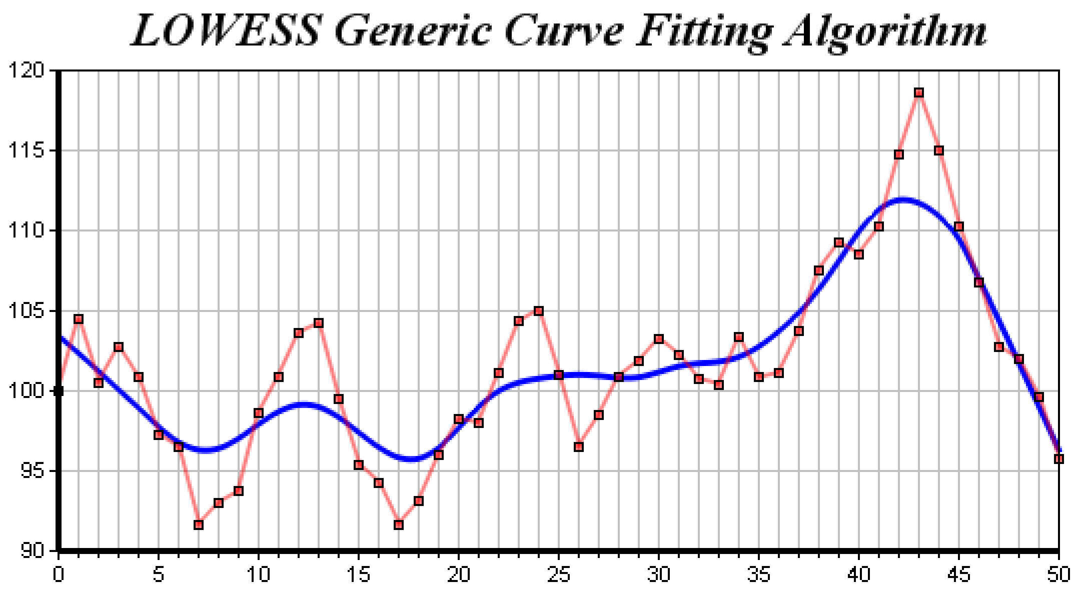
4. Line Charts

4.1 Curve Fitting

This new feature will be available for Series of Chart With Axes, inclunding Line, Area, Scatter, Stock and Bar charts: curve fitting. This is an enhancement filed in bugzilla: https://bugs.eclipse.org/bugs/show_bug.cgi?id=87823

4.1.1 Definition

Here is an example of curve fitting chart:



The blue line shows the general trend of the data. The idea is to find a mathematical model representing the data. This is usually done by mathematical regression and there are different algorithms available. For instance one approach is the u ser providing a parameterized function that he believes represents the data, the algorithm will compute the best approximation of the parameters based on the data points. Alternatively, it is possible the user does not assume any particular function. Several algorithms are available, the example shows the LOWESS algorithm.

The goals of curve fitting are numerous: once a curve can be computed, it is possible to seek its area, maxima, minima, tangentes, peak values, etc…

Specification

BIRT Chart will use the LOWESS algorithm, without parameterized function provided by the user. The curve will be drawn over the line chart in a color chosen by the user. No characteristics will be computed from the curve.

4.1.2 API

A new interface will be added

```
public interface FittingCurve
{
          LineAttributes getLineAttributes();
          void setLineAttributes(LineAttributes value);
          Label getLabel();
          void setLabel(Label value);
          Anchor getLabelPosition();
          void setLabelPosition(Anchor value);
          void unsetLabelPosition();
}
```

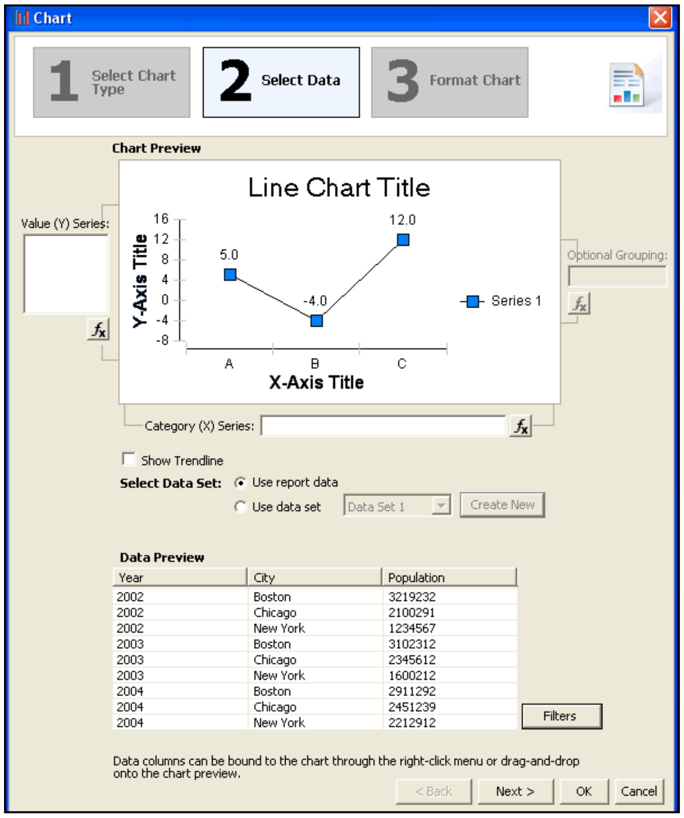
New methods will be added to the LineSeries, BarSeries, and StockSeries interface:

```
public interface LineSeries extends Series
```

```
    {
            …
            setFittingCurve( FittingCurve );
            FittingCurve getFittingCurve( );
            …
    }
```

4.1.3 Chart Builder

The Chart Builder UI will provide a checkbox to show the fitting curve, and a property sheet to set its attributes.



Year	City	Population
2002	Boston	3219232
2002	Chicago	2100291
2002	New York	1234567
2003	Boston	3102312
2003	Chicago	2345612
2003	New York	1600212
2004	Boston	2911292
2004	Chicago	2451239
2004	New York	2212912

Data columns can be bound to the chart through the right-click menu or drag-and-drop onto the chart preview.

4.2 Line coloring from Palette

4.2.1 Bugzilla Entries

https://bugs.eclipse.org/bugs/show_bug.cgi?id=100406

4.2.2 Feature Description

Allow the line color for a given serie to follow the palette colors when series grouping is applied. The user will have the choice to choose a fixed color or use the palette colors.

4.2.3 API Change

TBD

5. Meter/Dial Charts

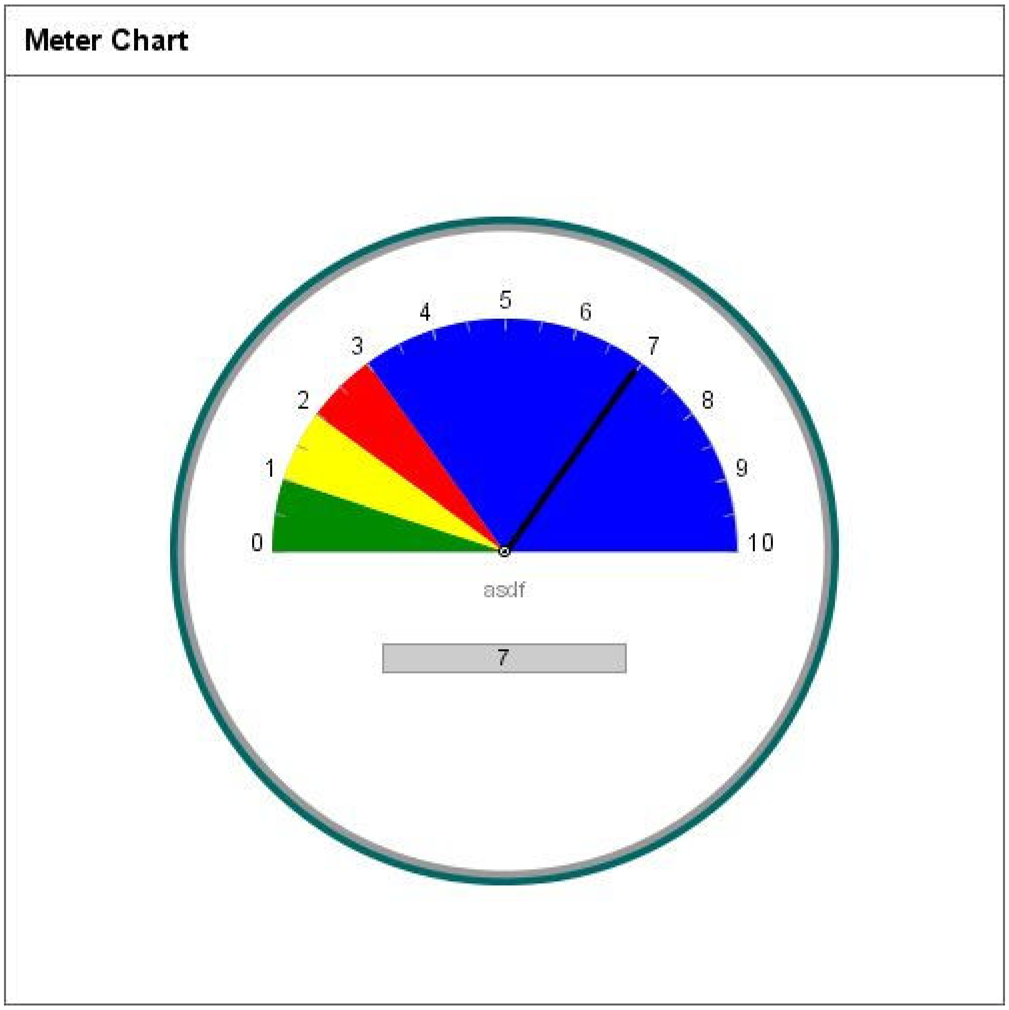
5.1 Bugzilla Entries

This is a new Chart type, part of the "chart without axis" family. This is an enhancement filed in bugzilla: https://bugs.eclipse.org/bugs/show_bug.cgi?id=101646

5.2 Definition

"Meter charts are useful for displaying the current value of a certain measurement. The value is indicated by a pointer and the scale of the meter dial. The background of the meter can be divided into regions with different colors, each representing a range of measurement values. The meanings of regions can be displayed in tooltips as the mouse pointer moves over them. Example of meter charts or dial graphs can be gauges, speedometers, or clocks." (Bugzilla entry)

Here is an example of a meter/dial chart:



The following are the basic components of a meter chart.

- Title
- Dial
- Legend

5.3 Title

The title section is similar to the title sections of other charts

5.4 Dial

A meter chart can be composed of several dials. A dial has the following components:

- Regions
- Needle
- Label
- Major Grid
- Minor Grid
- Scale
- Start Angle
- Stop Angle
- Radius

Regions

A region can span a specified area of a dial. A label can be associated with a region. In some rendering devices the label should be displayed as a tooltip. The user has the ability to format the label as well. A color can be associated with a region. The area of the region is specified by an inner radius, outer radius, start value and end value. An outline is also associated with a region.

Needle

A needle shows the value of the dial. A needle can support different end decorator styles such as arrow head, circle, none. A color can be associated with a needle that will also indicate the legend item color. A line style can also be associated with the needle to indicate the thickness of the needle and dash pattern of the needle.

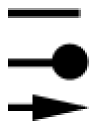


Figure 1 needle decorators

Label

The label is the text that identifies the dial that will be shown in the legend section.

Major Grid

Major Grid defines the cosmetic attributes of the major tick grid marks.

Minor Grid

Minor Grid defines the cosmetic attributes of the minor tick grid marks.

Scale

Specifies scale information such as minimum value, maximum value, step value, minor unit value and scale unit.

Start Angle

Start angle specifies the angle in degrees of the start position of the dial.

Stop Angle

Stop angle specifies the angle in degrees of the end position of the dial

Radius

Radius defines the radius of the dial.

5.5 Legend

A meter chart can consists of more than one dial. Only one needle is associated with a dial. A legend visually shows the numeric value of a needle. Each legend item represents a needle value. The label of the dial is displayed with the color of the needle and a text label that indicates the value of the needle.

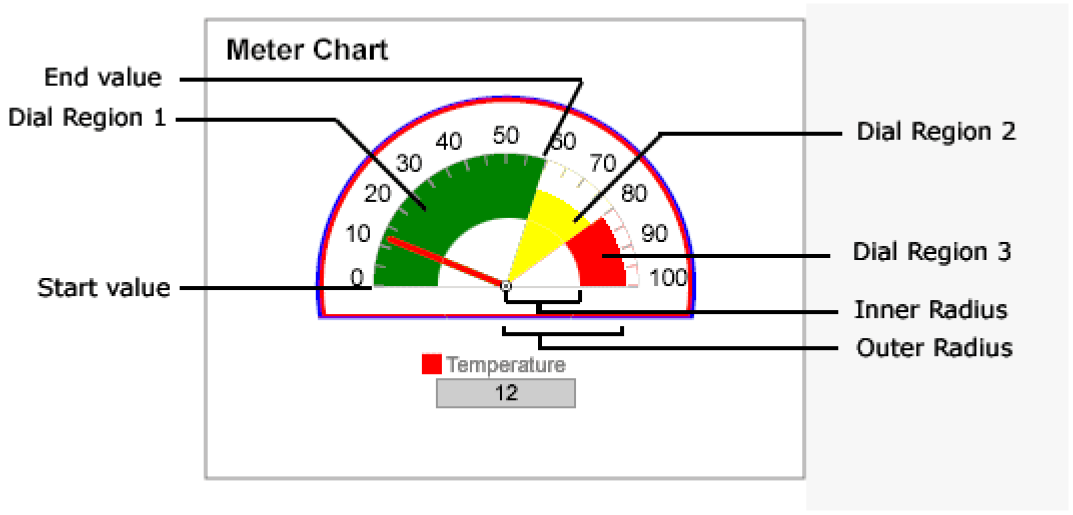


Figure 2 - dial chart that contains one dial and three dial ranges.

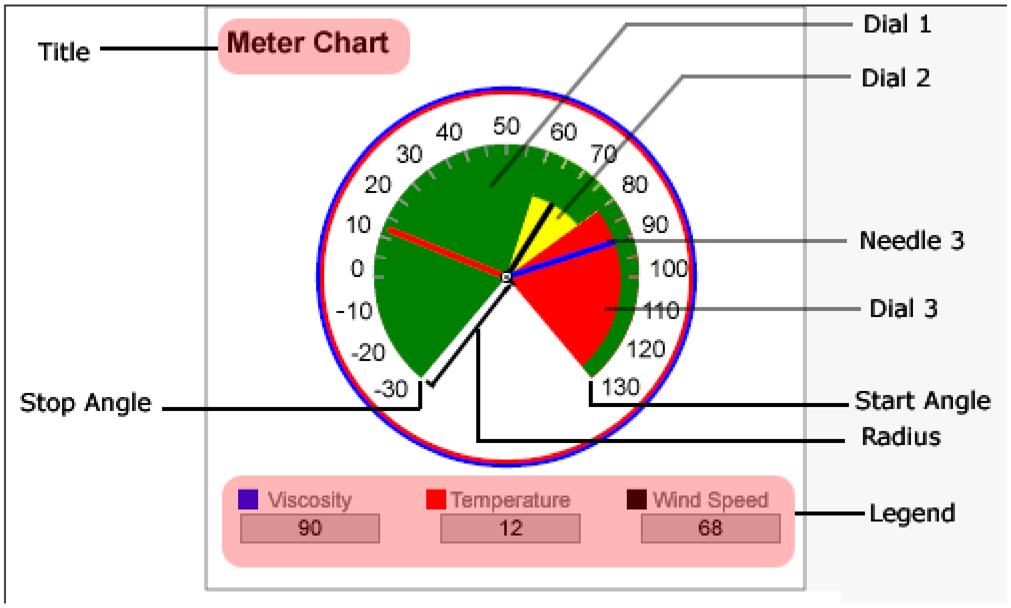


Figure 3 - dial chart that contains three dials. Each dial has one region.

5.6 API – Schema change

Two new elements should be added to the component.xsd schema.

- Dial
- DialRegion
- Needle

```
      <xsd:complexType name="Dial">
            <xsd:annotation>
                  <xsd:documentation xml:lang="en">
                  This type defines the basic elements that are expected in a dial
chart. This can further be extended for special dial chart types.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:sequence>
                  <xsd:element name="StartAngle" type="xsd:double" minOccurs="0">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies the start angle of the dial.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="StopAngle" type="xsd:double" minOccurs="0">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies the stop angle of the dial.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="Radius" type="xsd:double" minOccurs="0">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies the radius of the dial.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element                                name="LineAttributes"
type="attribute:LineAttributes">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies the border line style.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="Label" type="Label">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              A label instance to hold attributes for axis labels.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element                               name="FormatSpecifier"
type="attribute:FormatSpecifier" minOccurs="0">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies the formatting for dial labels.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element  name="DialRanges"  type="DialRange"  minOccurs="0"
maxOccurs="unbounded">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Defines a set of areas for a range of values within a
dial displayed as filled sections extending across the dial between the start and
end positions.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="MajorGrid" type="Grid">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Defines the major grid associated with the dial.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="MinorGrid" type="Grid">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Defines the minor grid associated with the dial.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="Scale" type="Scale">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Defines the scale for the dial.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
            </xsd:sequence>
      </xsd:complexType>


      <xsd:complexType name="Needle">
            <xsd:annotation>
                  <xsd:documentation xml:lang="en">
                  This type defines a needle of a dial.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:sequence>
                  <xsd:element                                name="LineAttributes"
type="attribute:LineAttributes">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specify the line properties for the needle.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="Decorator" type="attribute:LineDecorator">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specify the line decorator for the needle.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
            </xsd:sequence>
      </xsd:complexType>

      <xsd:complexType name="DialRegion">
            <xsd:annotation>
                  <xsd:documentation xml:lang="en">
                  This type defines a dial area. It is intended for use as a region
associated with an Axis.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:sequence>
                  <xsd:element name="Outline" type="attribute:LineAttributes">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specify the outline properties for the region.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="Fill" type="attribute:Fill">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specify the background color for the dial region.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="StartValue" type="data:DataElement">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Defines where this area starts relative to the dial
scale.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="EndValue" type="data:DataElement">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Defines where this area is ends relative to the dial
scale.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="Label" type="Label">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies the label associated with this area.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="LabelAnchor" type="attribute:Anchor">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies where the label associated with this area is
to be positioned within respect to the dial range itself.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element                                name="FormatSpecifier"
type="attribute:FormatSpecifier" minOccurs="0">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies the formatting for dial range labels.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="InnerRadius" type="xsd:double " minOccurs="0">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies the percentage value of the inner radius of
the dial range.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="OuterRadius" type="xsd:double " minOccurs="0">
                        <xsd:annotation>
                              <xsd:documentation xml:lang="en">
                              Specifies the percentage value of the outer radius of
the dial range.
                              </xsd:documentation>
                        </xsd:annotation>
                  </xsd:element>

            </xsd:sequence>
      </xsd:complexType>
```

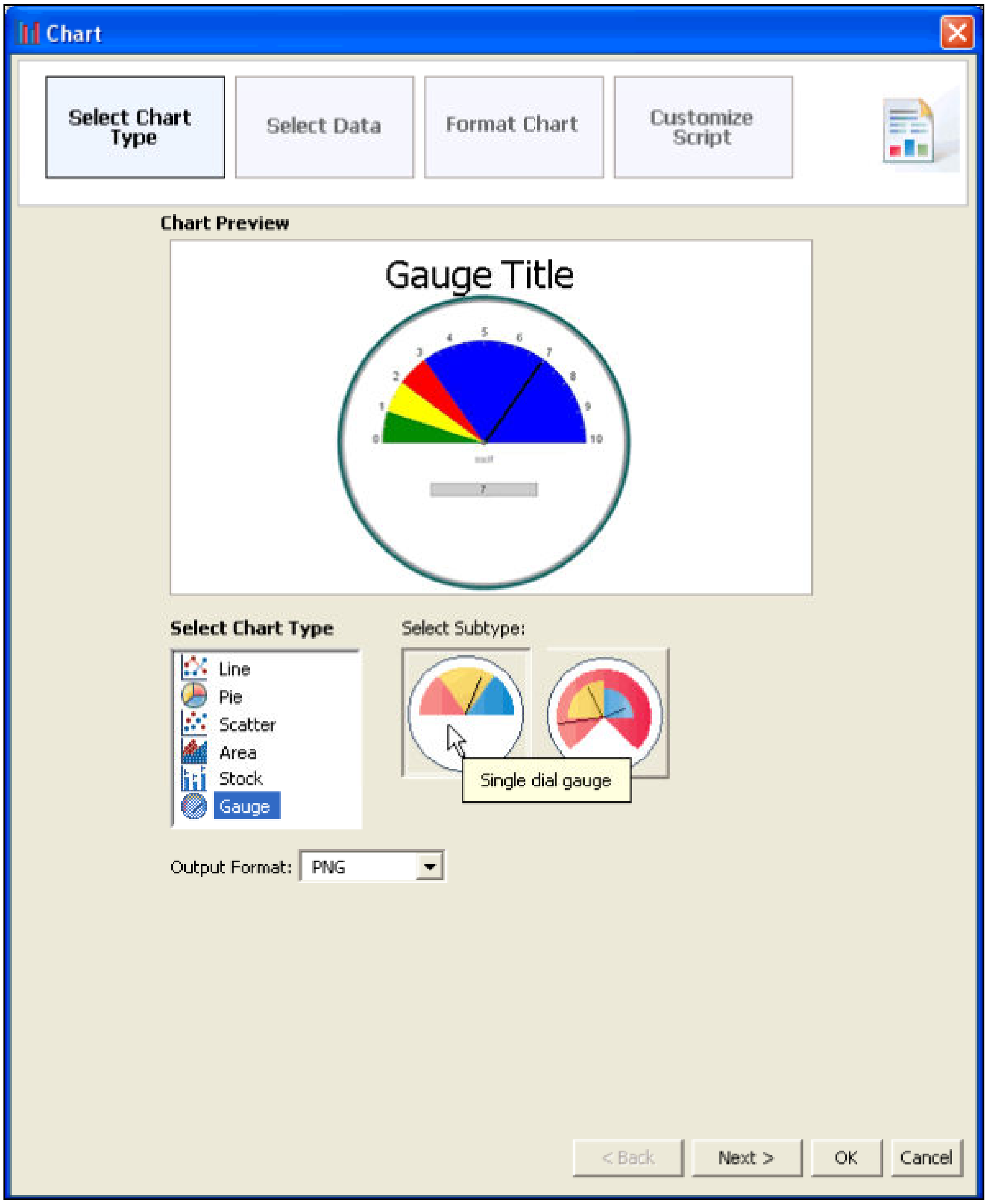
A new simple type should be added to the attribute.xsd schema.

- LineDecorator

```
      <xsd:simpleType name="LineDecorator">
            <xsd:annotation>
                  <xsd:documentation xml:lang="en">
                  This type represents the possible line head decorators.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:restriction base="xsd:string">
                  <xsd:enumeration value="Arrow"/>
                  <xsd:enumeration value="Circle"/>
                  <xsd:enumeration value="None"/>
            </xsd:restriction>
      </xsd:simpleType>
```

5.7 Chart Builder

The chart builder for the meter chart should be similar to the chart builder for pie charts. As a comparison we can compare a dial in a meter chart with a pie section in a pie chart. Therefore the basic workflow of the chart builder for the meter chart is similar to building a pie chart. Most of the chart builder panels are the same between the meter chart and pie chart. Only the chart build panels under the data section and the attributes sections are different. Notice that the following screen captures show the new panels for building a meter chart. Also notice that some of these panels are based on the old chart builder user interface.



Output Format: PNG

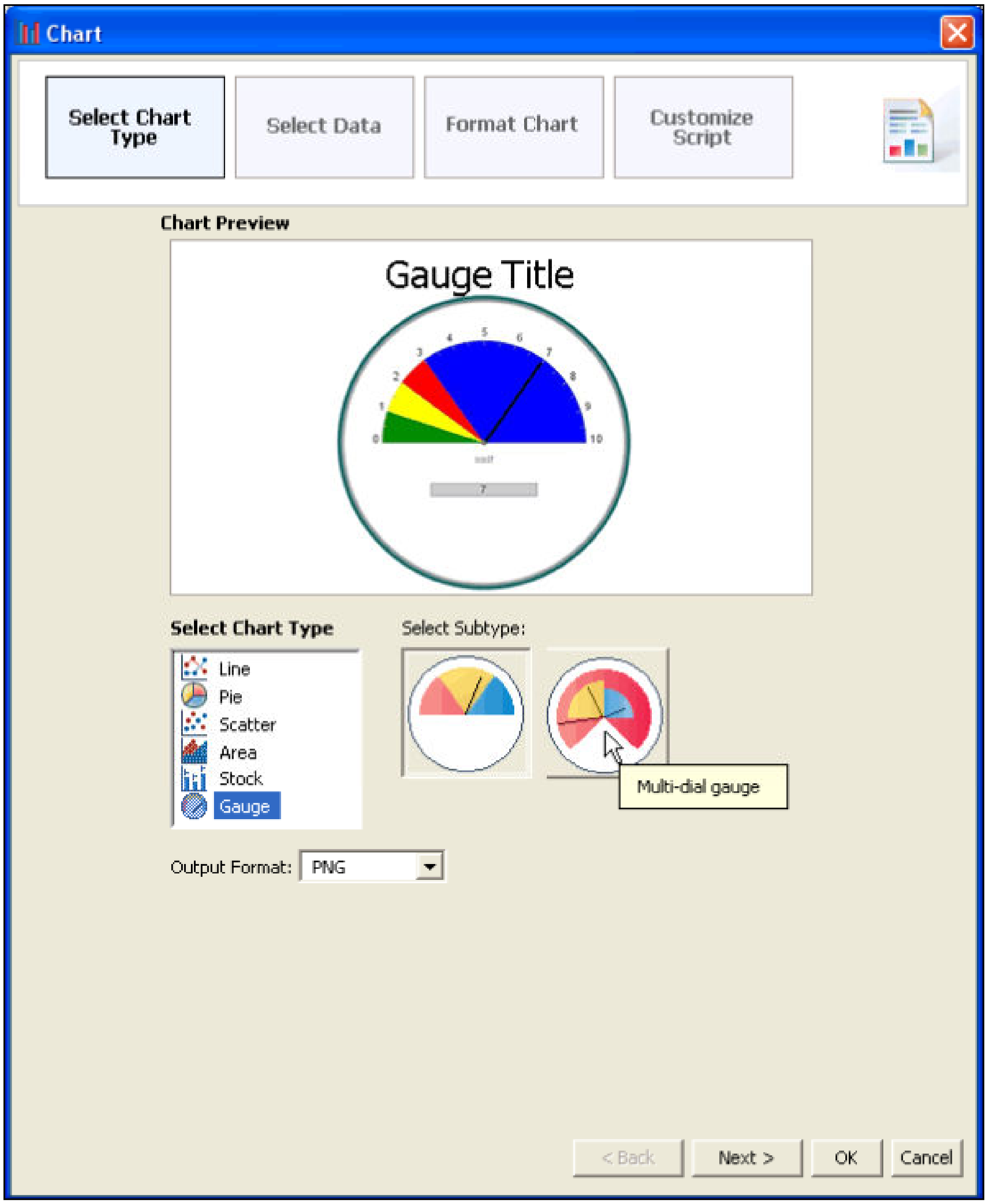
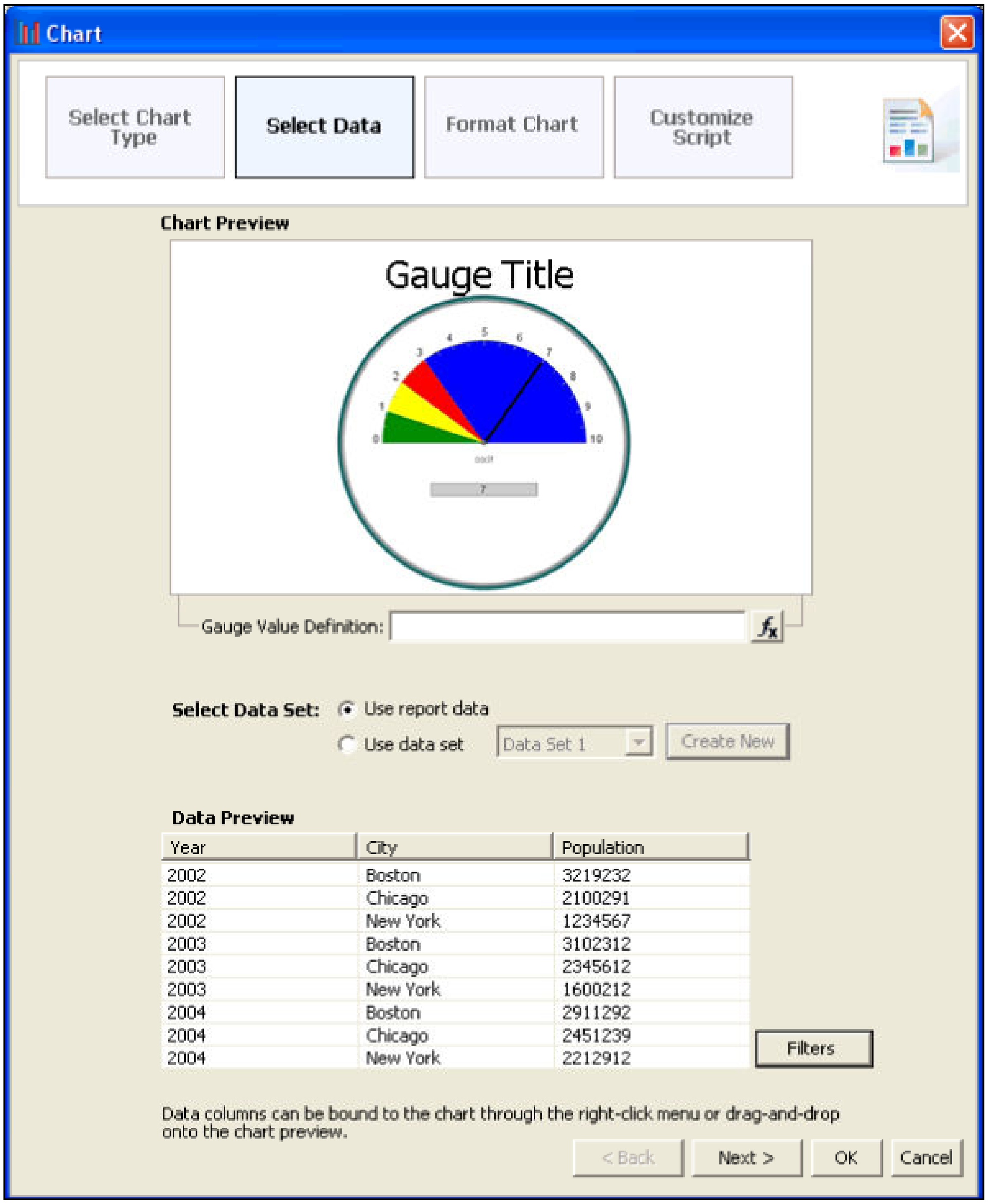


Chart
Select Chart Type
Select Data
Format Chart
Customize Script
Chart Preview
Gauge Title
0
1
2
3
4
5
6
7
8
9
10
7
Select Chart Type
Line
Pie
Scatter
Area
Stock
Gauge
Select Subtype:
Multi-dial gauge
Output Format:
PNG
< Back
Next >
OK
Cancel

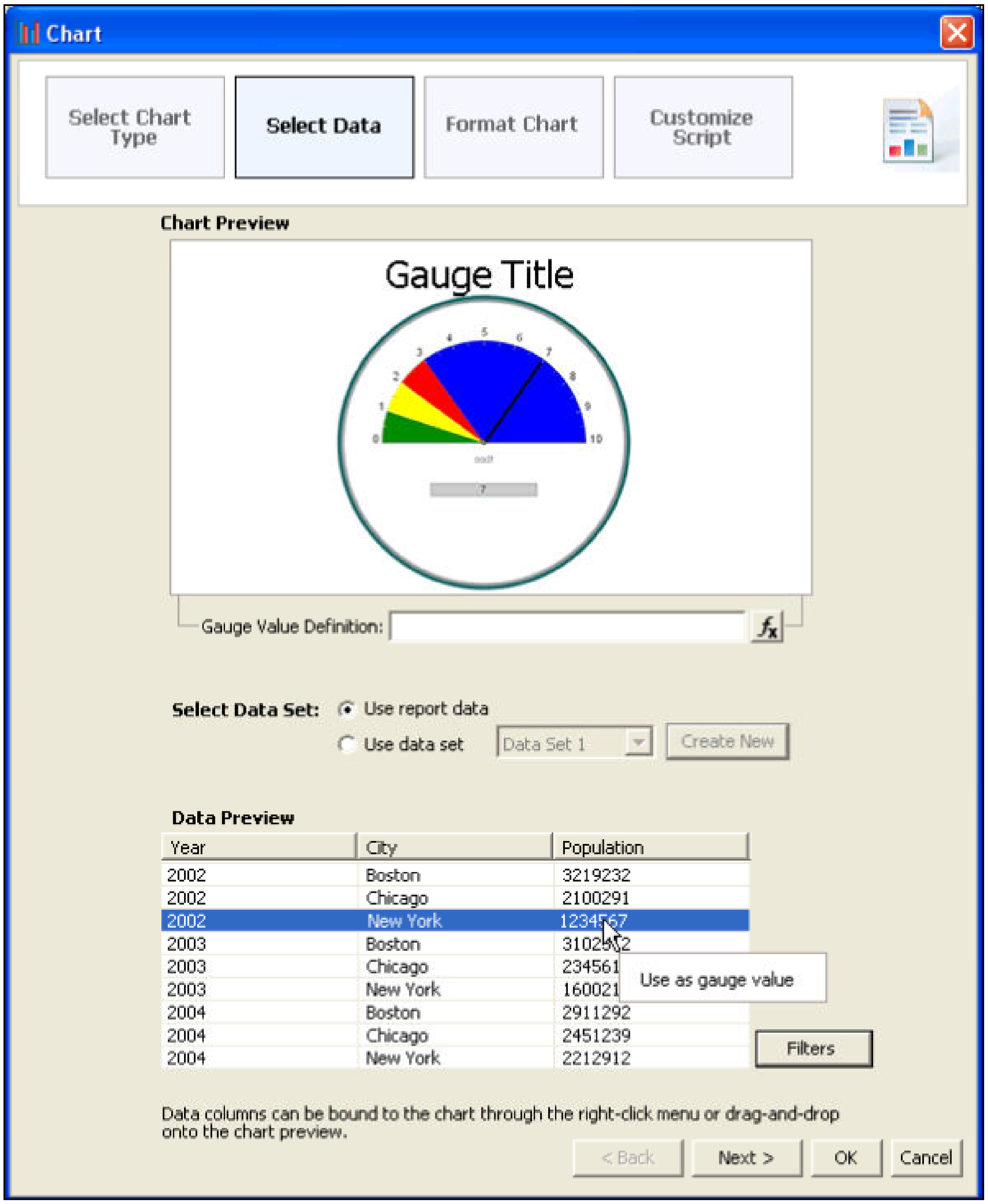


Select Data Set: [✓ Use report data] Use data set Data Set 1

Data Preview

Year	City	Population
2002	Boston	3219232
2002	Chicago	2100291
2002	New York	1234567
2003	Boston	3102312
2003	Chicago	2345612
2003	New York	1600212
2004	Boston	2911292
2004	Chicago	2451239
2004	New York	2212912

Data columns can be bound to the chart through the right-click menu or drag-and-drop onto the chart preview.



Select Data Set: [✓ Use report data] Use data set

Year	City	Population
2002	Boston	3219232
2002	Chicago	2100291
2002	New York	1234567
2003	Boston	3102512
2003	Chicago	234561
2003	New York	160021
2004	Boston	2911292
2004	Chicago	2451239
2004	New York	2212912

Data columns can be bound to the chart through the right-click menu or drag-and-drop onto the chart preview.

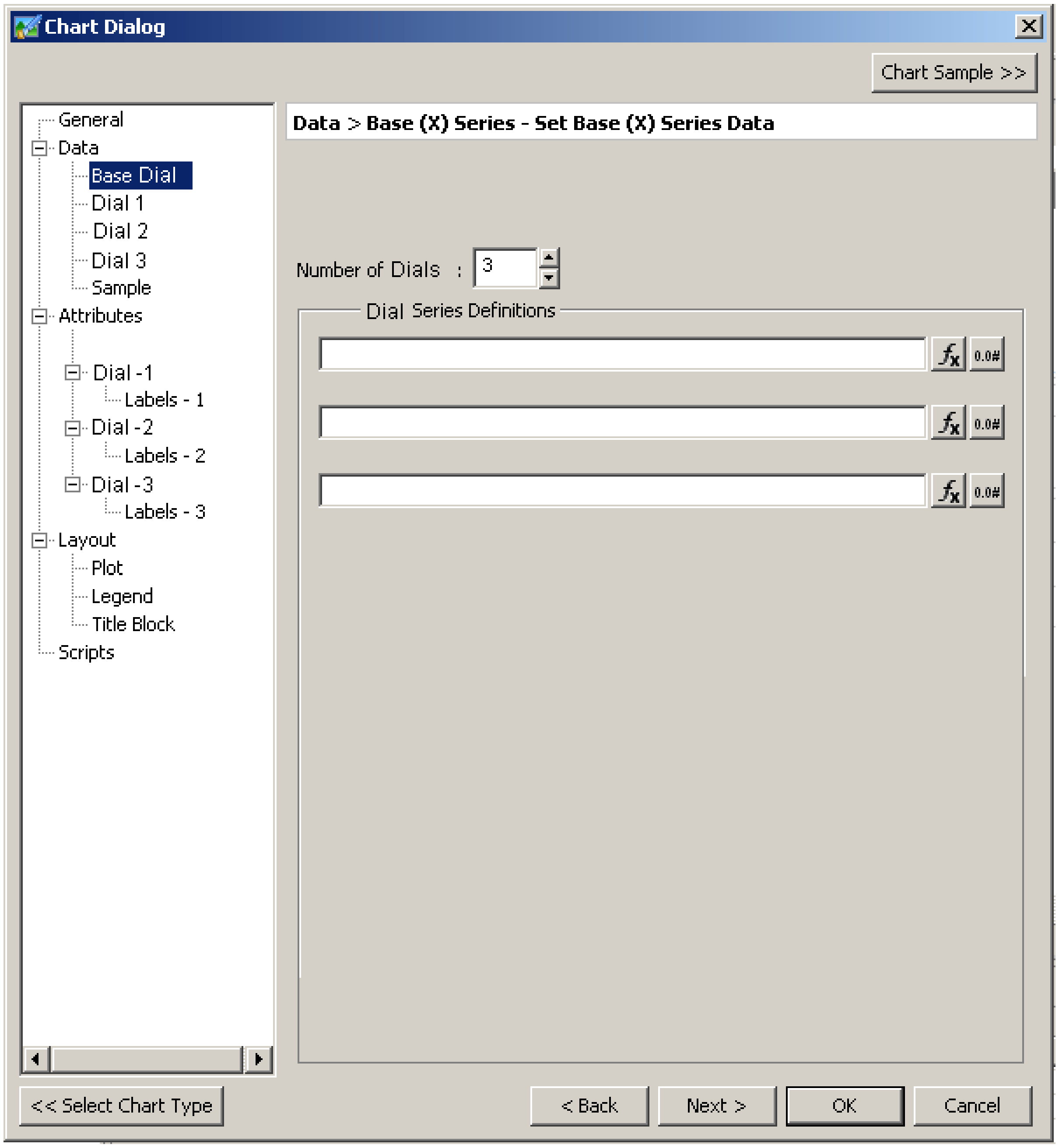


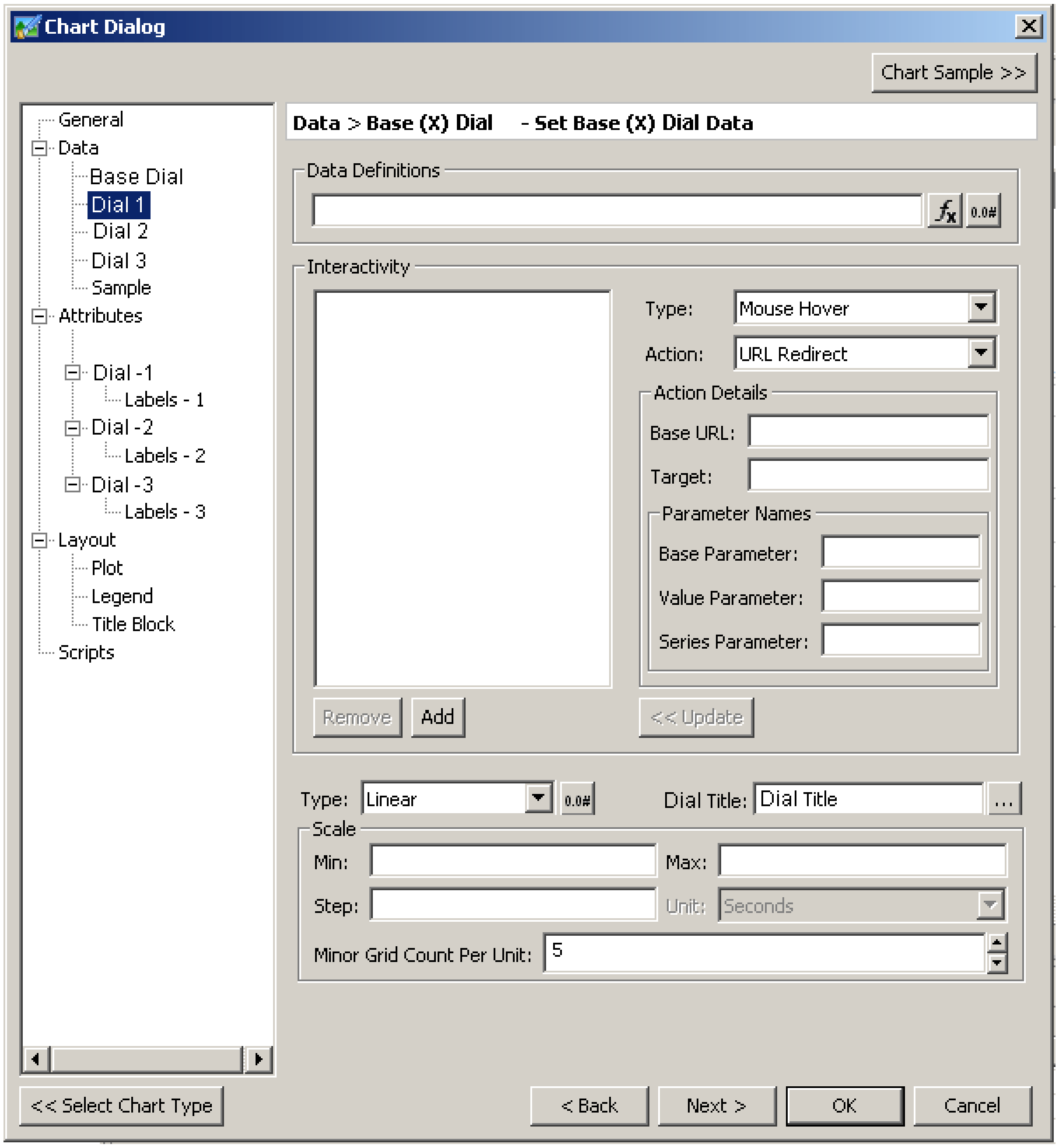
Chart Dialog

Chart Sample >>

Data > Base (X) Series - Set Base (X) Series Data

Number of Dials : 3

Dial Series Definitions



Data > Base (X) Dial - Set Base (X) Dial Data

Data Definitions

Interactivity

Type: Mouse Hover

Action: URL Redirect

Action Details

Base URL:

Target:

Parameter Names

Base Parameter:

Value Parameter:

Series Parameter:

Type: Linear

Dial Title: Dial Title

Scale

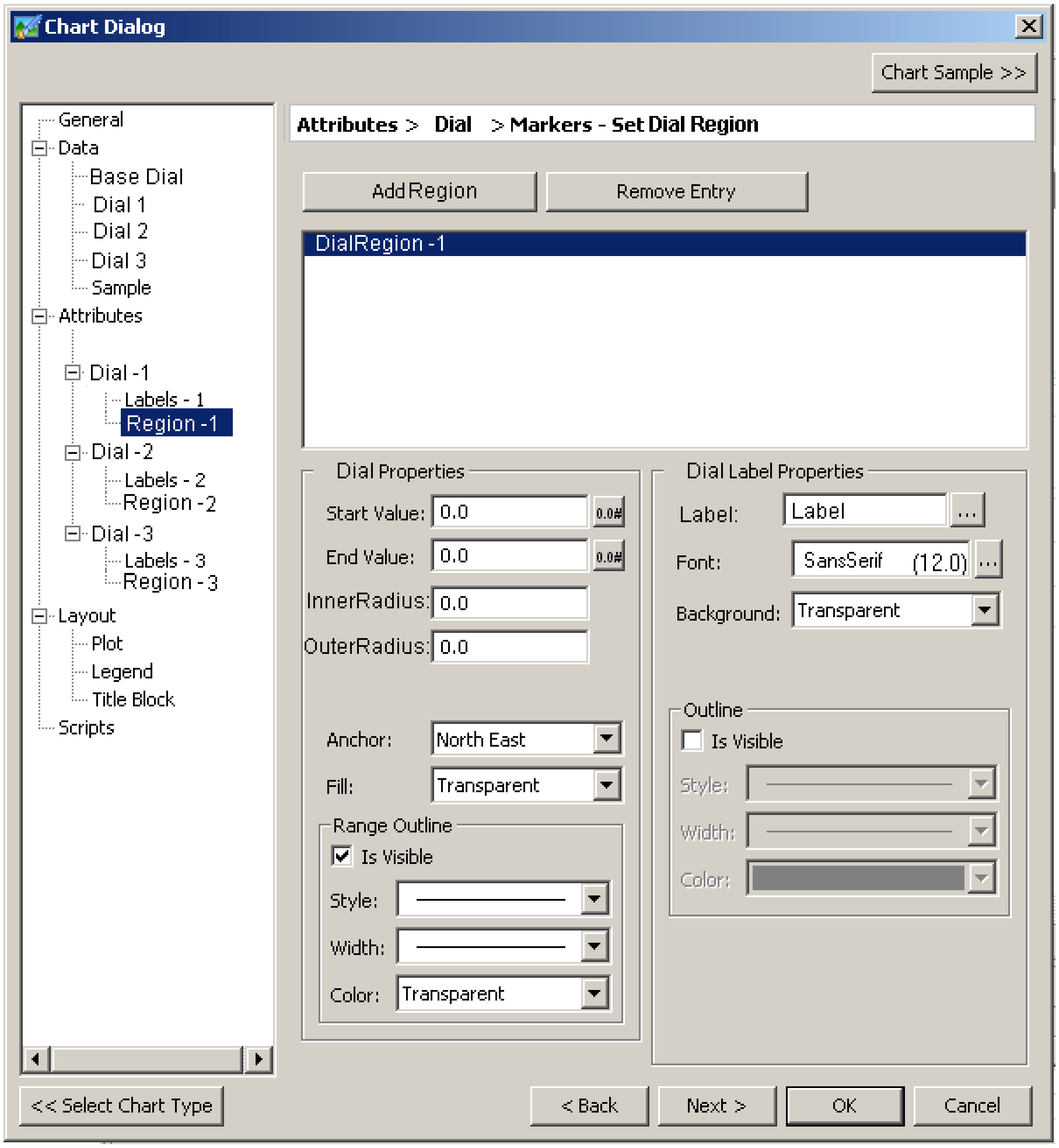
Min:

Max:

Step:

Unit: Seconds

Minor Grid Count Per Unit: 5



Attributes > Dial > Markers - Set Dial Region

DialRegion -1

Dial Properties

Start Value: 0.0

End Value: 0.0

InnerRadius: 0.0

OuterRadius: 0.0

Anchor: North East

Fill: Transparent

Range Outline

[✓] Is Visible

Style:

Width:

Color: Transparent

Dial Label Properties

Label: Label

Font: SansSerif (12.0)

Background: Transparent

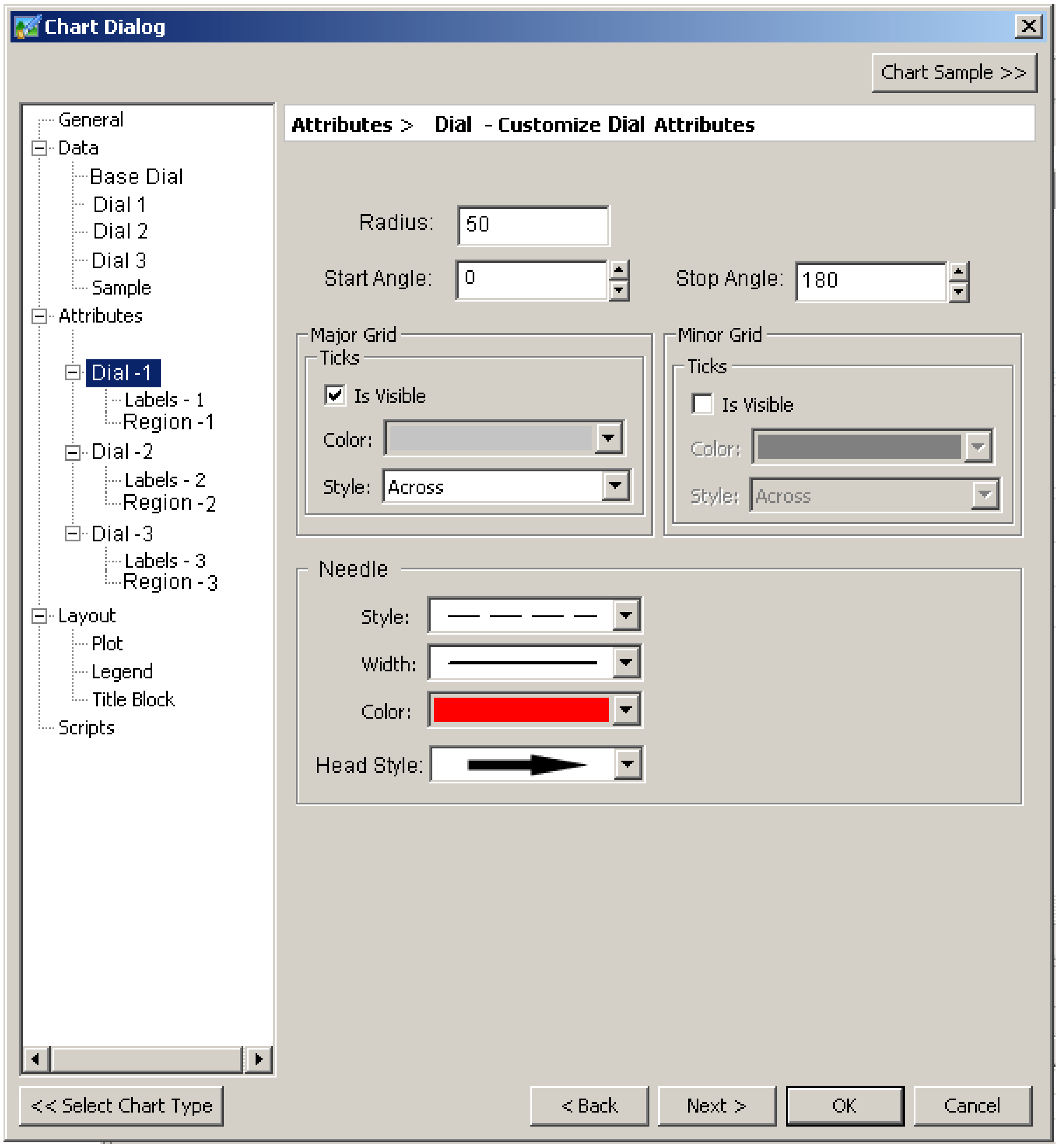
Outline

[ ] Is Visible

Style:

Width:

Color:



Attributes > Dial - Customize Dial Attributes

Radius: 50

Start Angle: 0

Stop Angle: 180

Major Grid

Ticks

[✓] Is Visible

Color:

Style: Across

Minor Grid

Ticks

[ ] Is Visible

Color:

Style: Across

Needle

Style:

Width:

Color: 🟥

Head Style:

6. Area Charts

6.1 There are three enhancements in bugzilla requiring this new chart type:

101648,101651,101407

6.2 Definition

Area charts are very similar to Line Charts, the only difference being that the area between the X axis and the line is filled in color. The painted areas can be set as transparent or opaque (transparent being the default for a 3D Area Chart, whereas opaque is the default for a 2D Area Chart).

Area charts benefit of course of all the Line Chart features. Let's only mention they can be stacked, and percent stacked as well, shown in 2D, 2D with depth or 3D.

6.3 API

The transparency of the Area is controlled on the Series interface level:

```
public interface Series extends EObject
{
…
        boolean isTranslucent();
        void setTranslucent(boolean value);
…
}
```

The AreaSeries simply extends the LineSeries interface, without any new method:

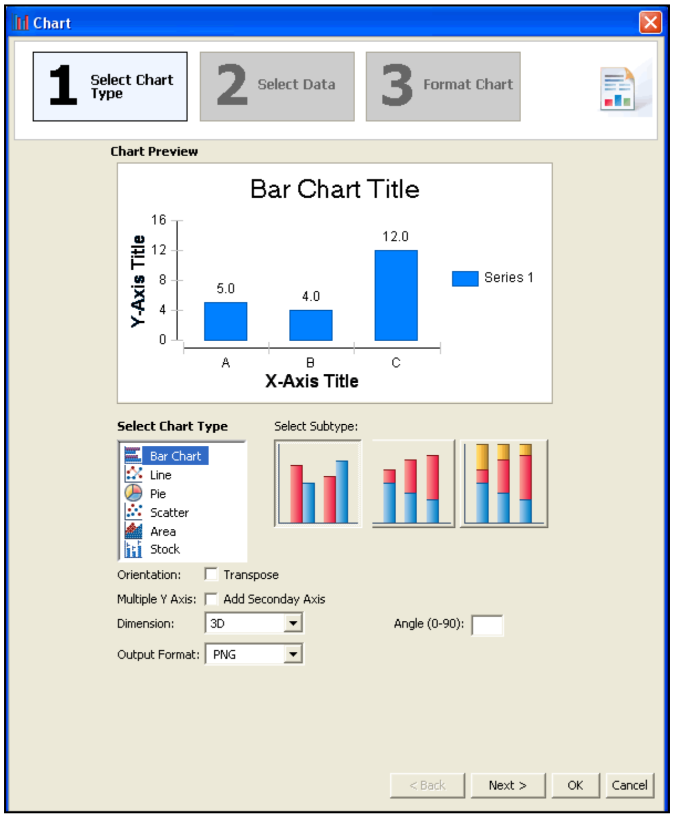
```
public interface AreaSeries extends LineSeries
{
}
```

Note that by default no marker is shown for the AreaSeries

6.4 Chart Builder

A new type will be added in the first screen of the builder for Area Charts. It can be declined in stacked and percent stacked as well.

The transparency of the area will be set in the Y series attribute page.



Orientation: Transpose

Multiple Y Axis: Add Seconday Axis

Dimension: 3D

Angle (0-90):

Output Format: PNG