

BIRT Chart Types Enhancements

Functional Specifications

Draft 5: November 1st, 2005

Abstract

*This document describes the **new** 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.
4	August 18, 2005	David Michonneau	Modified schema of Dial Charts and detailed the rendering possibilities
5	November 1st, 2005	David Michonneau	Updated schemas for 3D model, Pie Chart, Dial Chart. Updated screenshots.

Table of Contents

1. Introduction	2
2. 3D Charts	3
2.1 3D Definition	3
2.2 Chart types in 3D	4
2.3 API	4
2.4 Chart Builder	7
3. Pie Chart	7
3.1 Explosion	7
3.2 Minimum Slice	8
3.3 Pie Chart Ratio	9
3.4 Pie Chart Builder	11
4. Line Charts	12
4.1 Curve Fitting	12
4.2 Line coloring from Palette	13
5. Meter/Dial Charts	14
5.1 Bugzilla Entries	14
5.2 Definition	15
5.3 Title	15
5.4 Dial	15
5.5 Legend	17
5.6 Rendering Behavior	17
5.7 Mockups	17
5.8 API – Schema change	18
6. Area Charts	30
6.1 There are three enhancements in bugzilla requiring this new chart type:	30
6.2 Definition	30
6.3 API	30
6.4 Chart Builder	30

1. Introduction

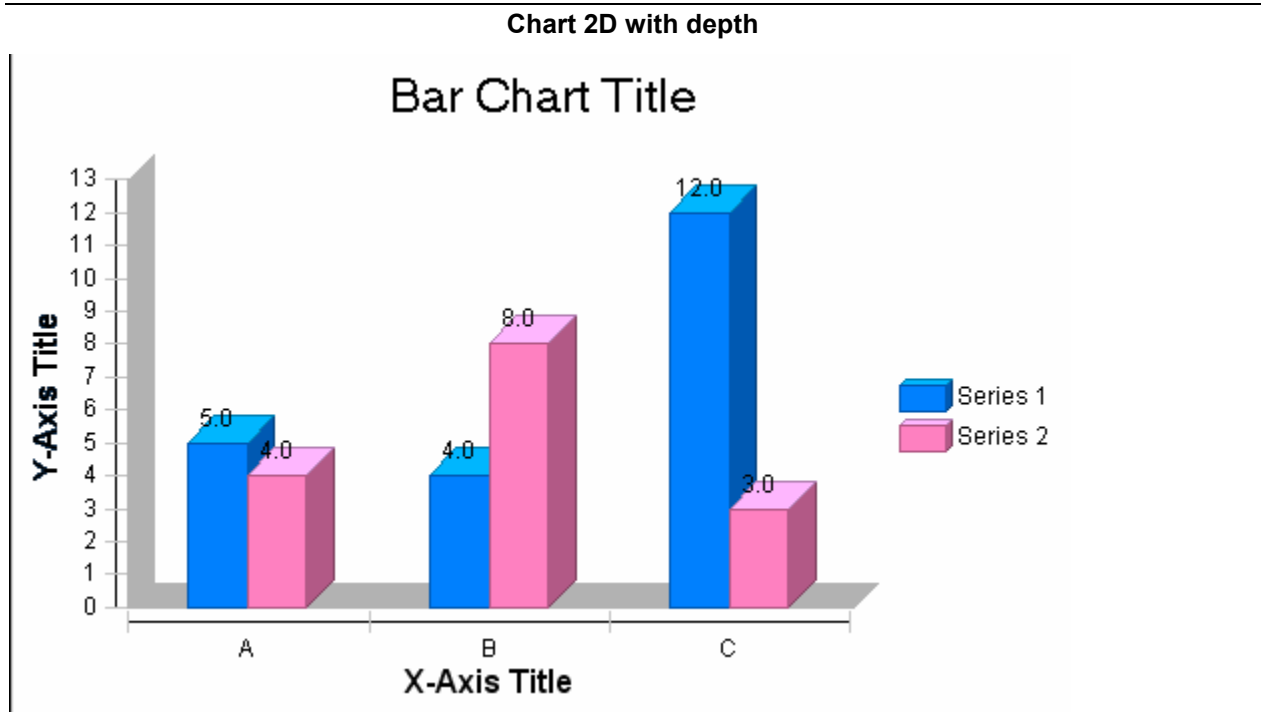
Chart types are a combination of series types and chart attributes. They can be divided into two categories: charts with axis and chart without axis (such as pie charts). New series types will be provided, as well as improvements of the existing ones. Some new chart attributes will also be added to support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

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

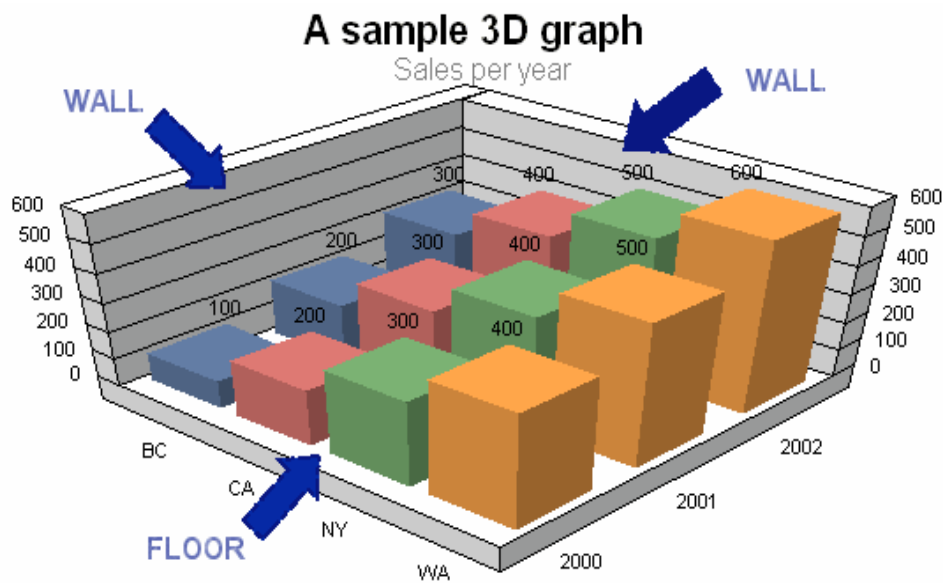
2. 3D Charts

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:



3D Chart



Copyright, 2004 - Actuate Corporation

The 3D chart not only displays the series with depth: each series 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

Combination charts (multiple Y axis) are not supported

Transposition is not supported

Subtypes are not supported (series will render one behind each other along the Z axis)

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/Rotation attribute

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

```
public interface ChartWithAxes extends Chart
{
    ...
    Rotation3D get3DRotation();
    void set3DRotation( Rotation3D rotationAngles);
    ...
}

// Specify the rotation with a list of angles along different axes.
public interface Rotation3D
{
    // returns a list of Angle3D in a specific order.
    Elist getAngles();
}

// specify the angle of rotation on a given axis
public interface Angle3D
{
    AngleType getType();
    void setType();
    double getAxisAngle();
    void setAxisAngle( double angleInDegrees );
}

// Defines the axis reference for an angle: X, Y, or Z axis
public final class AngleType
{
    public static final int X = 1;
    public static final int Y = 2;
    public static final int Z = 3;
}
```

Each of the rotation angle will be applied clockwise, and must be within the 0-90 range. The **Rotation3D** property is only effective for 3D charts and will be ignored otherwise.

The order in which the rotations are applied matter. For instance a 90 rotation on the X axis followed by a 90 rotation on the Y axis leads to a different transformation if applied in reverse order (if you think in rotation matrixes, it's because the matrix product is not commutative).

2.3.3 Z Axis

A Z Axis can be associated in the X-Axis of the ChartWithAxes class. The following attributes and properties control its behavior:

Type	Supported for Z Axis	Special Meaning
Title	✓	-
Subtitle	-	-
TitlePosition	✓	-
AssociatedAxes	-	-

Type	Supported for Z Axis	Special Meaning
SeriesDefinition	✓	Can be used for 3D scatter charts. Otherwise provides label values on the Z Axis for non-numerical dataset, or Z axis position for numerical datasets. The SeriesDefinition is optional.
GapWidth	-	-
Orientation	-	-
LineAttributes	✓	-
Label	✓	-
FormatSpecifier	✓	-
LabelPosition	✓	-
Staggered	✓	-
MarkerLines	-	-
MarkerRanges	-	-
MajorGrid	✓	-
MinorGrid	✓	-
Scale	✓	-
Origin	✓	Only min is supported
PrimaryAxis	✓	-
CategoryAxis	✓	-
Percent	-	-

2.3.4 Floor/Wall attributes

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

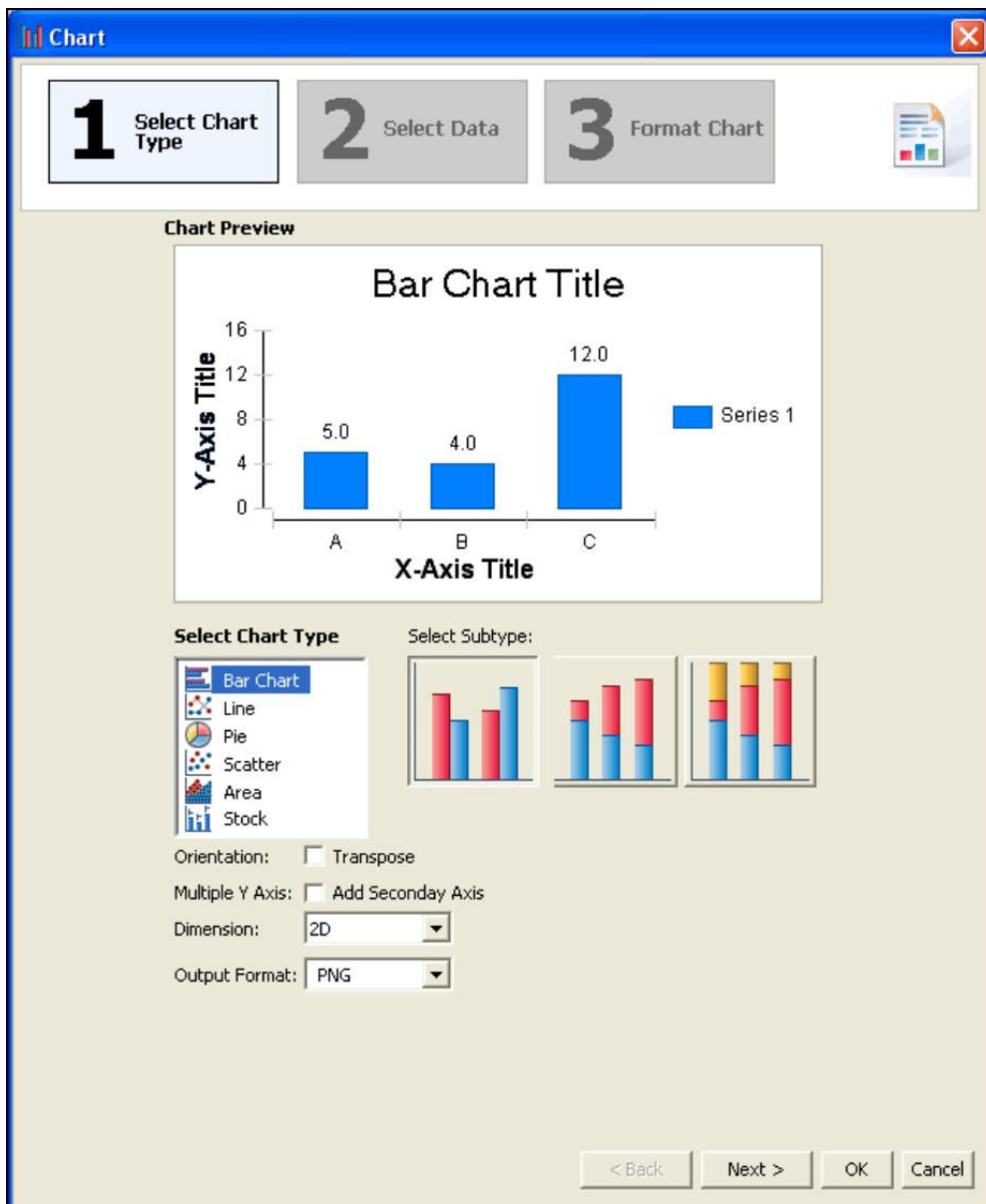
```

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.



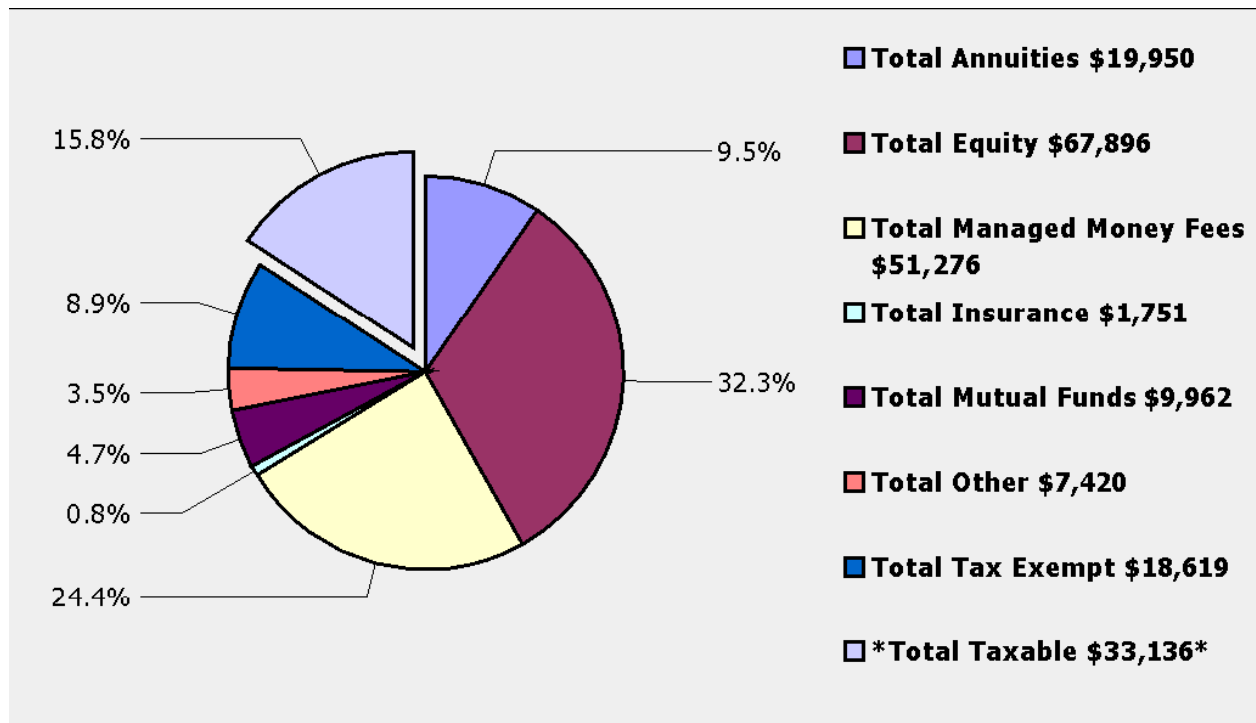
3. Pie Chart

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.

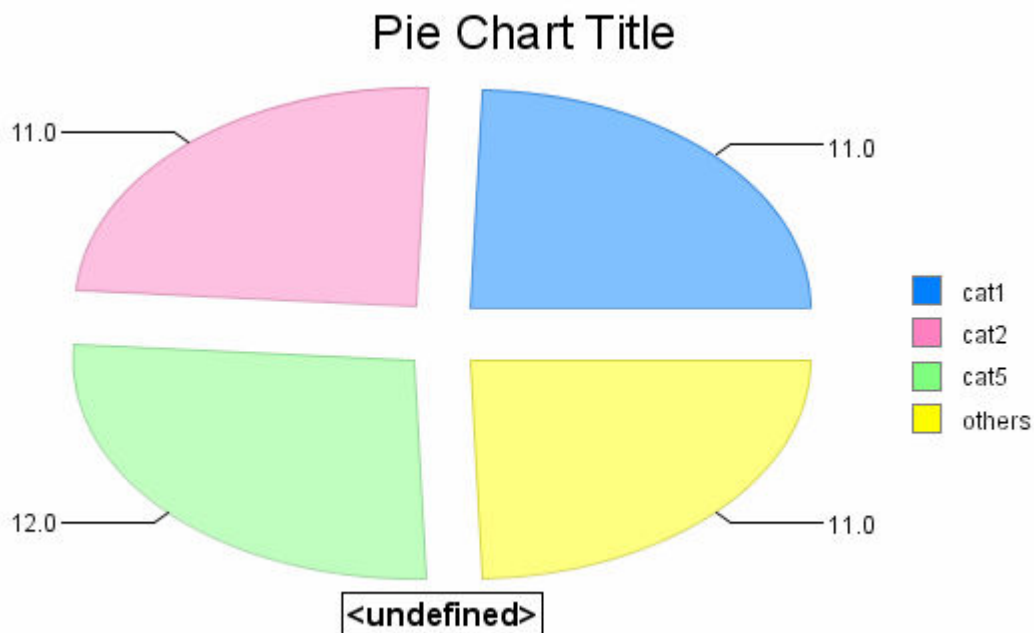
3.1.1 API

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 be 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 Ratio

3.3.1 Description

It is now possible to control the height/width ratio of a chart or have it stretched automatically. A ratio of 1 means the chart is rendered as a circle. A ratio of 0 means that the pie is automatically stretched based on the chart size, to fill the maximum possible area inside the plot.

3.3.2 API

```
<xsd:complexType name="PieSeries">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      This is a Series type that holds data for Pie Charts.
    </xsd:documentation>
  </xsd:annotation>
</xsd:complexType>
```


```

</xsd:annotation>
<xsd:complexContent>
  <xsd:extension base="component:Series">
    <xsd:sequence>
      <xsd:element name="Explosion" type="xsd:int" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            Specifies the 'Explosion' value to be used while displaying the pie slices.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="ExplosionExpression" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Specifies an expression to determine if the explosion will be applied to each
slice.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Title" type="component:Label">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            Holds the properties for a series title.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TitlePosition" type="attribute:Position">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            Holds the position property for a series title.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="LeaderLineAttributes" type="attribute:LineAttributes">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            Holds the attributes for leader lines.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="LeaderLineStyle" type="attribute:LeaderLineStyle">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            Specifies how the leader lines are to be shown.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="LeaderLineLength" type="attribute:Percentage">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            Length for the leader lines. Used only if style is 'FIXED_LENGTH'.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="SliceOutline" type="attribute:ColorDefinition" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation xml:lang="en">
            Defines the color to be used for the slice outline.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Ratio" type="xsd:double" default="0">
        <xsd:annotation>
          <xsd:documentation>Defines the ratio(height/width) of the oval, 1 means it's a circle, 0 means stretch
automatically.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:extension>

```

```
</xsd:complexContent>
</xsd:complexType>
```

3.4 Pie Chart Builder

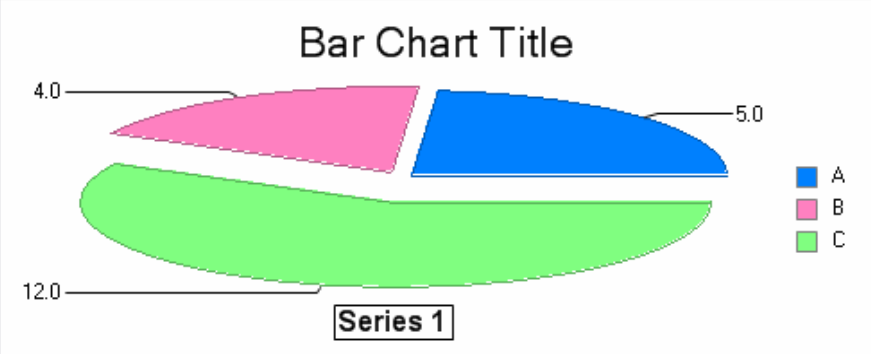

Chart Builder

Select Chart Type

Select Data

Format Chart

Chart Preview



Slice Size Definition:

Series 1

+

f_x 0.0#

Category Definition:

f_x 0.0#

Explode Slice When:

f_x 0.0#

Explosion Distance:

10

Minimum Slice:

Value

0.0

%

Select Data Set:

☒ Use report data
 ☐ Use data set

Create New

Data Preview

Filters

Parameters

To bind a data column to the chart, use the right-click menu or drag a column onto the chart preview.

<< Back

Next >>

OK

Cancel

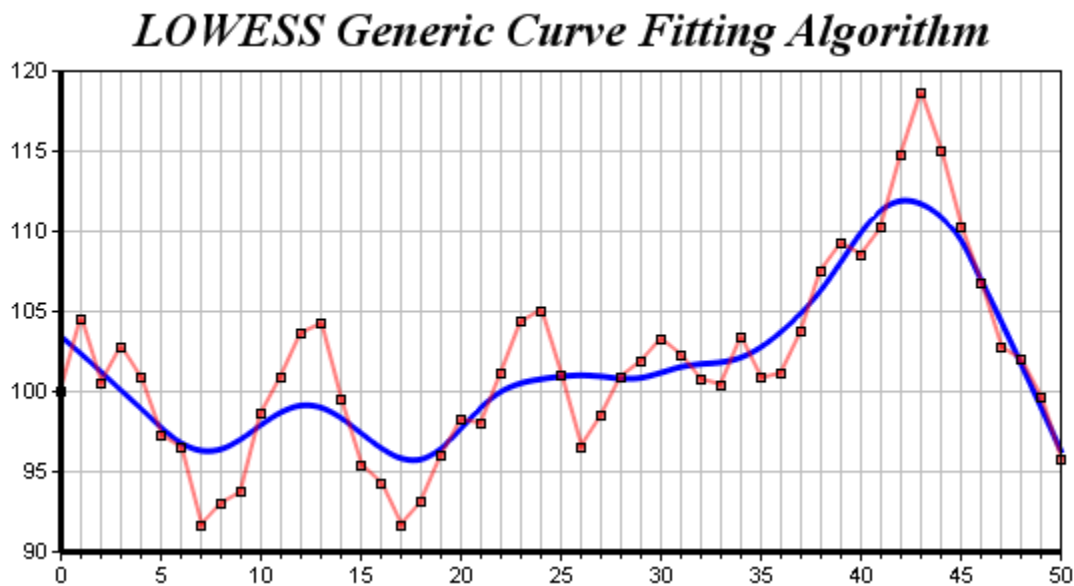
4. Line Charts

4.1 Curve Fitting

This new feature will be available for Series of Chart With Axes, including 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 user 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:

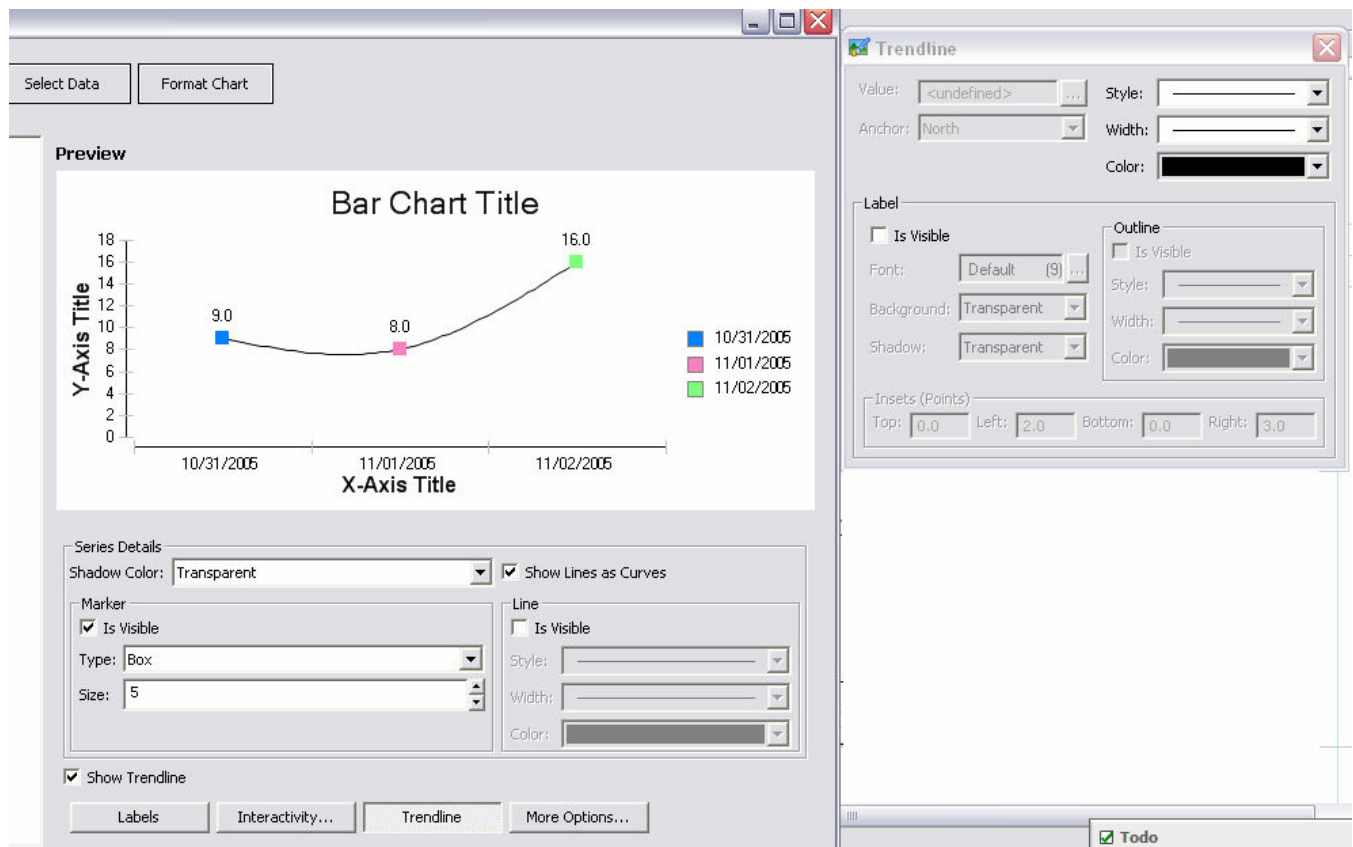
```

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.



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

```
<!-- Line Series -->
<xsd:complexType name="LineSeries">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      This is a Series type that holds data for Line Charts.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="component:Series">
      <xsd:sequence>
        <xsd:element name="Marker" type="attribute:Marker" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              Specifies the marker to be used for displaying the data point on the line in the chart.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="LineAttributes" type="attribute:LineAttributes" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              Specifies the attributes for the line used to represent this series.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="PaletteLineColor" type="xsd:boolean">
          <xsd:annotation>
            <xsd:documentation>Indicates if use the series palette color to draw the line instead of the color in
LineAttributes</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Curve" type="xsd:boolean">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              Indicates whether the line segments joining data points in the series are to be drawn as curves or as
straight lines.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="ShadowColor" type="attribute:ColorDefinition">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              Specifies the color to be used for the shadow.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

5. Meter/Dial Charts

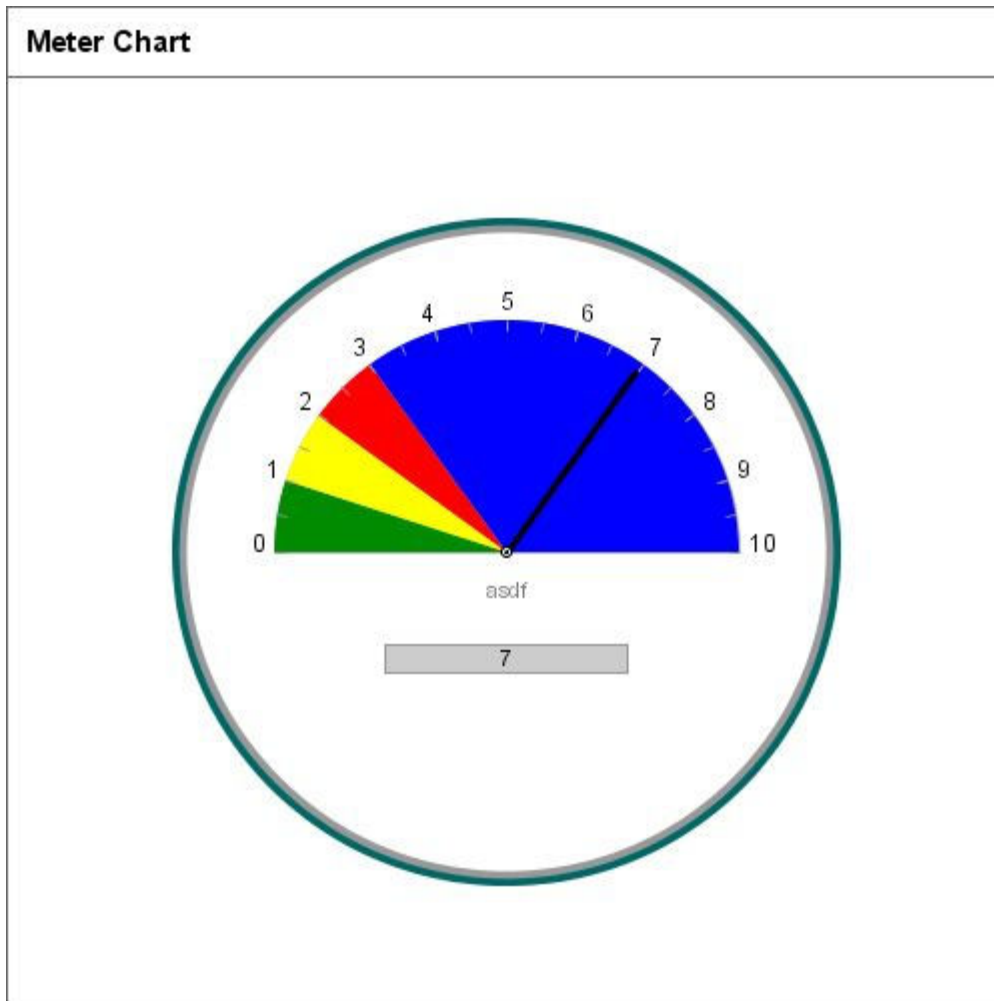
5.1 Bugzilla Entries

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.

5.6 Rendering Behavior

Here are some considerations regarding the chart output:

Dials can be superimposed, or drawn below each other. That's a model property.

The scale numbers are drawn for each dial, on the half-circle based on the radius of the region with the biggest outer radius. It is the responsibility of the designer to make sure he does not define dials that end up with overlapping numbers. It's of course possible to hide them for some dials.

Each Dial can define several regions of different colors and radius.

If the start and stop angle difference is less than 180 degrees (taking the max and min over all dials if superimposed), the chart is rendered inside a half disc. Otherwise the chart is rendered inside a full disc.

Dials can define similar region to simulate a single dial with multiple needles.

5.7 Mockups

Some mockups hereafter that show some examples of charts:

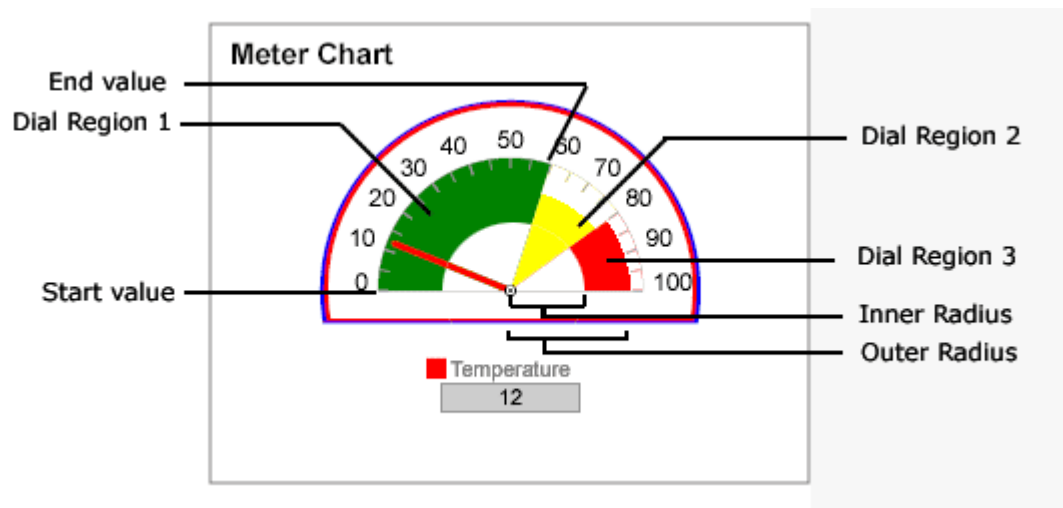


Figure 2 - dial chart that contains one dial and three dial regions.

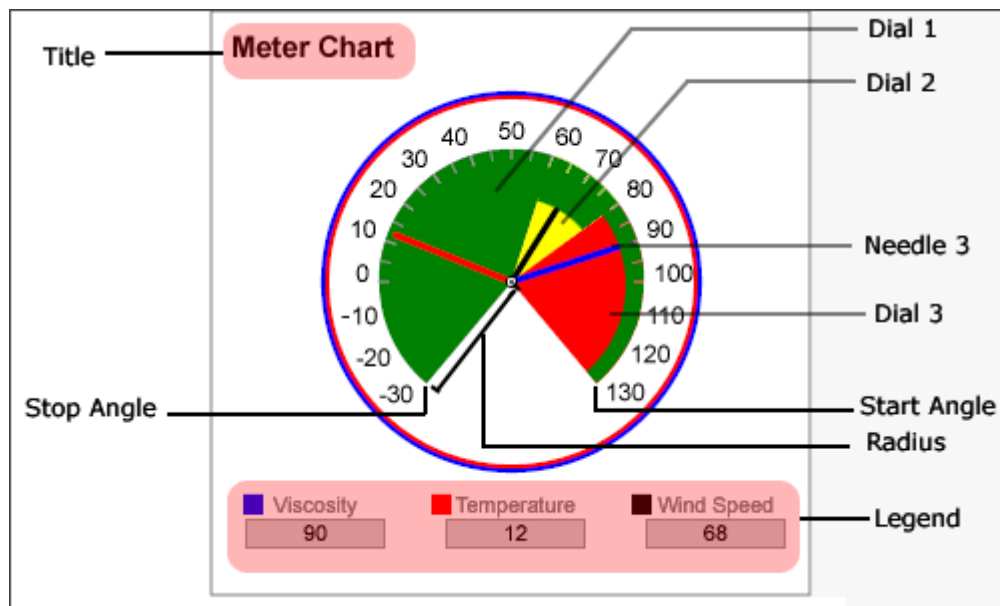


Figure 3 - dial chart that contains three dials. Each dial has one region.



Figure 4 - dial chart that contains three dials with identical regions.

5.8 API – Schema change

5.8.1 Model

A new element is added to model.xsd, extending the ChartWithoutAxes: DialChart

5.8.1.1 DialChart

```
<xsd:complexType name="DialChart">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      This is a dial chart
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="Chart">
      <xsd:sequence>
        <xsd:element name="DialsSuperImposition" type="xsd:boolean" default="true"
          minOccurs="0" maxOccurs="1">
            <xsd:annotation>
              <xsd:documentation xml:lang="en">
                If true, all dials are superimposed, otherwise they are rendered next to
                each other.
              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

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

5.8.2 Type

A new type of series is added to type.xsd: DialSeries

```
<xsd:complexType name="DialSeries">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      This is a Series type that holds data for Dial Charts.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="component:Series">
      <xsd:sequence>
        <xsd:element name="Dial" type="component:Dial" minOccurs="1" maxOccurs="1">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              Specifies the Dial for this series
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Needle" type="component:Needle" minOccurs="1" maxOccurs="1">
          <xsd:annotation>
            <xsd:documentation xml:lang="en">
              Defines the needle to be used in the Dial
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

5.8.3 Component

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

- Dial
- DialRegion
- Needle

5.8.3.1 Dial

```
<xsd:complexType name="Dial">
  <xsd:annotation>
    <xsd:documentation> 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>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>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>Specifies the radius of the dial. </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

```

<xsd:element name="LineAttributes" type="attribute:LineAttributes">
  <xsd:annotation>
    <xsd:documentation>Specifies the border line style.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Fill" type="attribute:Fill" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Specifies the background fill style.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="DialRegions" type="DialRegion" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>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>Defines the major grid associated with the dial.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="MinorGrid" type="Grid">
  <xsd:annotation>
    <xsd:documentation>Defines the minor grid associated with the dial.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Scale" type="Scale">
  <xsd:annotation>
    <xsd:documentation>Defines the scale for the dial.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Label" type="Label">
  <xsd:annotation>
    <xsd:documentation>Defines the properties for grid labels.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="FormatSpecifier" type="attribute:FormatSpecifier" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Format specifier for grid label.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

```

5.8.3.2 Needle

```

<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:sequence>
</xsd:complexType>

```

5.8.3.3 DialRegion

```

<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:extension base="MarkerRange">
    <xsd:sequence>
      <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 region.
          </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 region.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:extension>
</xsd:sequence>
</xsd:complexType>

```

5.8.4 Attribute

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

5.8.4.1 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>

```

5.8.5 Layout

5.8.5.1 Legend

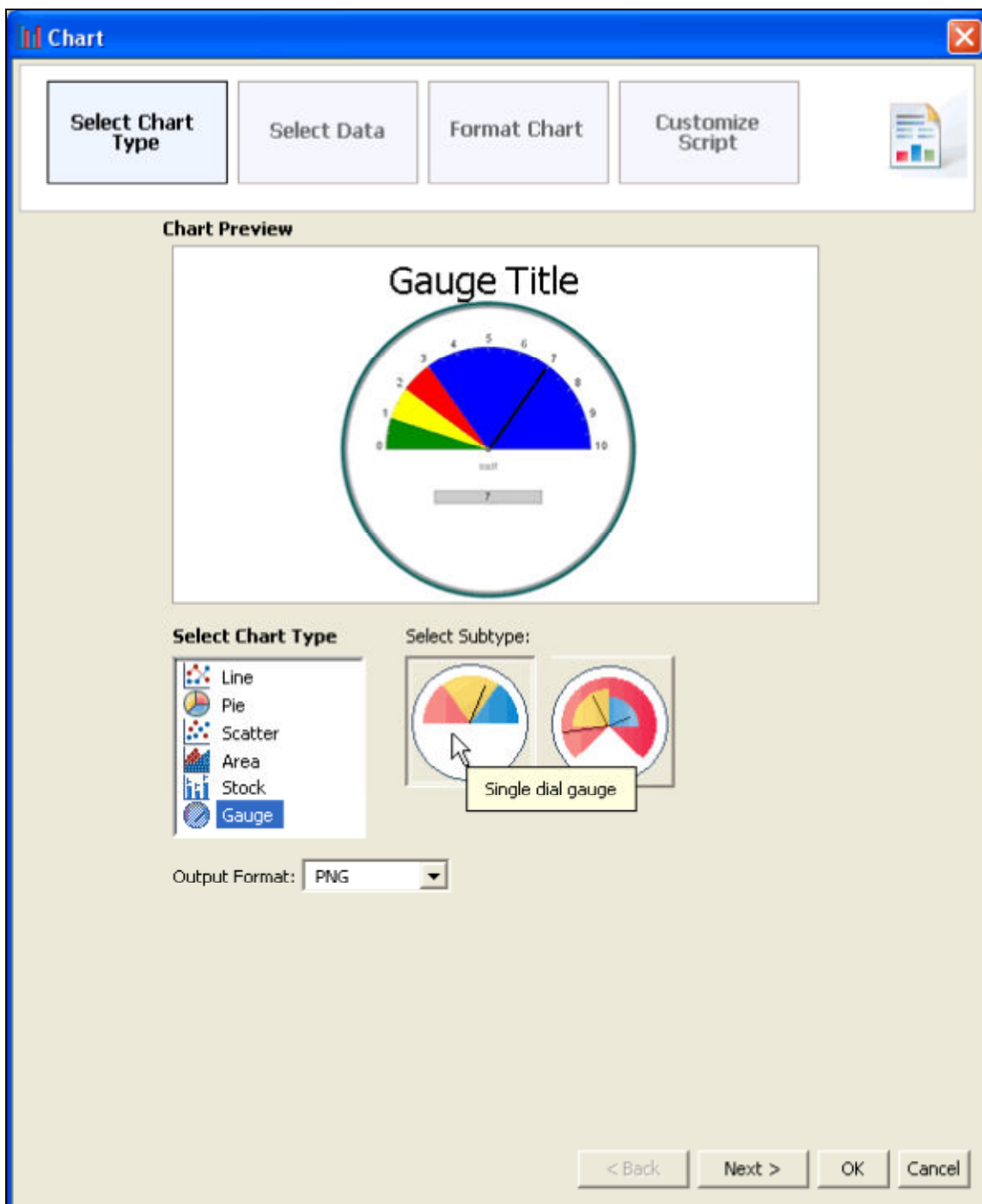
The legend has new options to display the value of each needle, and possibly their total. It can be used for other charts too, but it only takes the first value of the category.

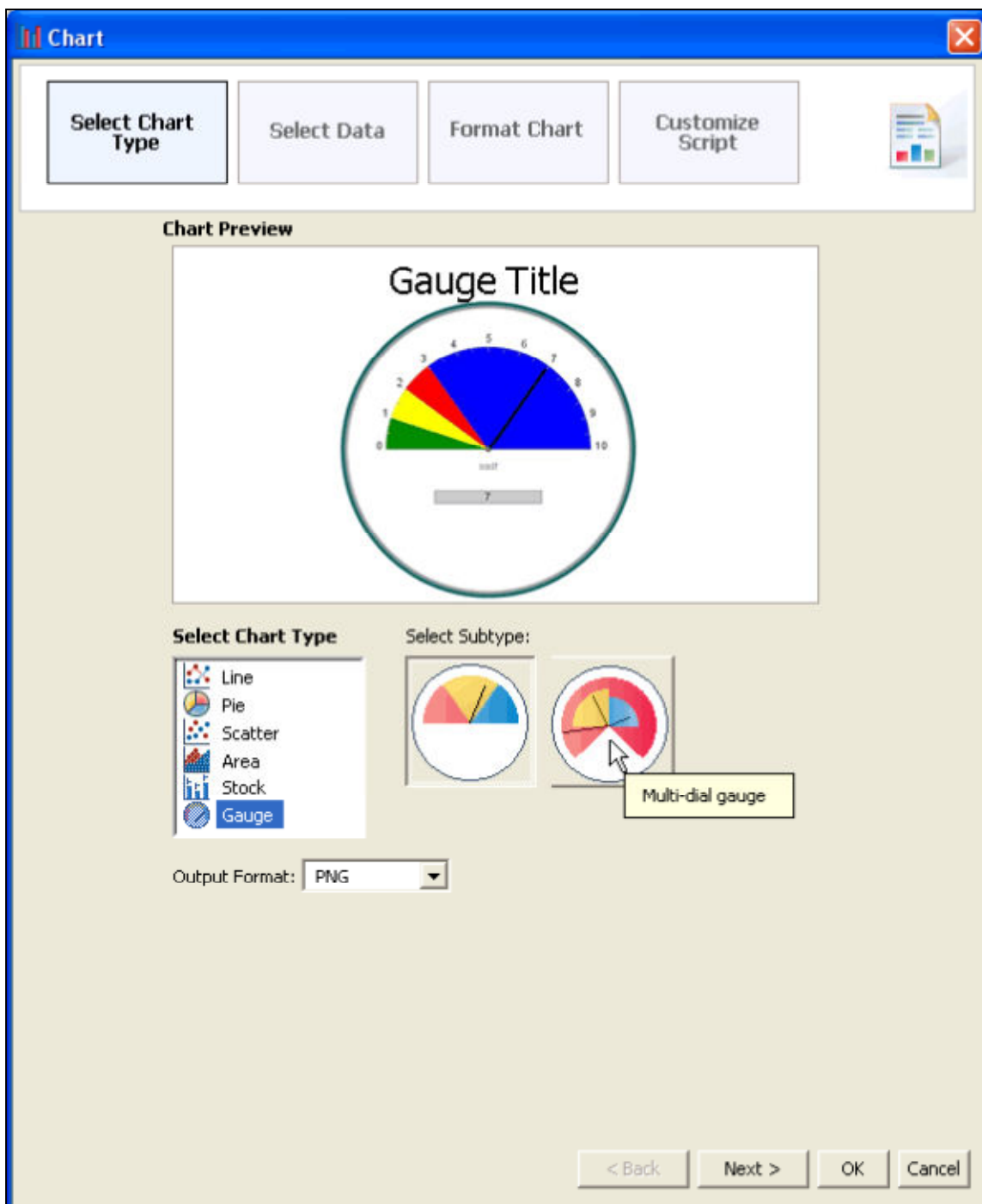
```
<xsd:complexType name="Legend">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The Legend in a chart.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="Block">
      <xsd:sequence>
        ...
        <xsd:element name="ShowValue" type="xsd:boolean" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Specifies if show legend item value.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="ShowPercent" type="xsd:boolean" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Specifies if show legend item value as percentile.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="ShowTotal" type="xsd:boolean" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Specifies if show legend item value total.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

5.8.6 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.

There is a maximum of two dials per chart in the chart builder, but the engine supports any number of dials.





Chart

Select Chart Type

Select Data

Format Chart

Customize Script

Chart Preview

Gauge Title

Gauge Value Definition:

Select Data Set:

☒ Use report data
☐ Use data set

Data Set 1

Create New

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

Filters

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

< Back

Next >

OK

Cancel

Chart

Select Chart Type


Select Data

Format Chart

Customize Script

Chart Preview

Gauge Title



Gauge Value Definition:

Select Data Set:

☒ Use report data
☐ Use data set

Data Set 1

Create New

Data Preview

Year	City	Population
2002	Boston	3219232
2002	Chicago	2100291
2002	New York	1234567
2003	Boston	3102912
2003	Chicago	234561
2003	New York	160021
2004	Boston	2911292
2004	Chicago	2451239
2004	New York	2212912

Use as gauge value

Filters

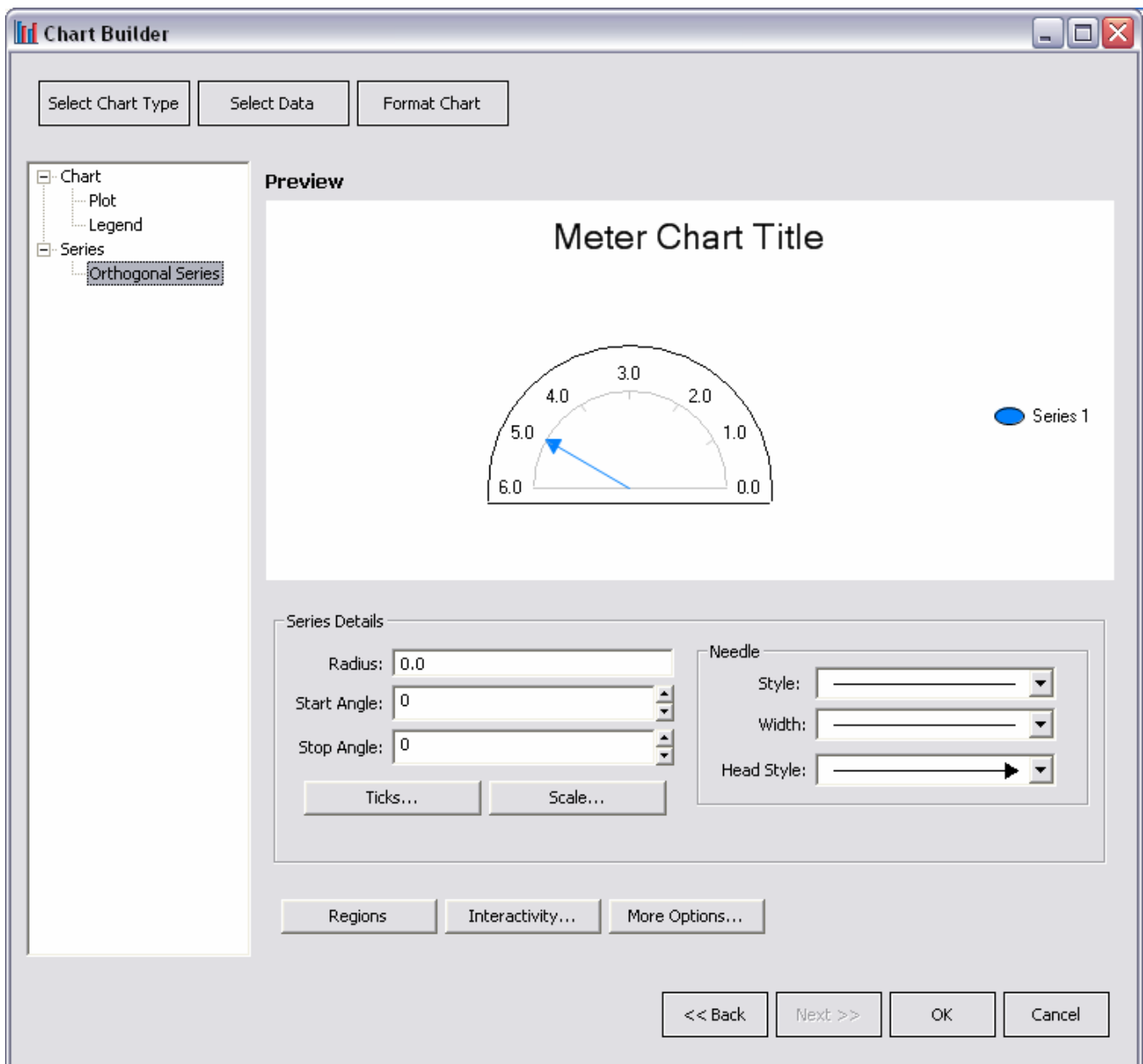
Data columns can be bound to the chart through the right-click menu or drag-and-drop onto the chart preview.


< Back

Next >

OK

Cancel





Add Region

Remove Entry

DialRegion - 1

Region Properties

Start Value: 0.0 0.0#

End Value: 0.0 0.0#

Inner Radius: 0.0

Outer Radius: 0.0

Fill: Transparent

Range Outline

Is Visible

Style:

Width:

Color:

Region Label Properties

Font: Default (9) ...

Background: Transparent


Outline

Is Visible

Style:

Width:

Color:



Scale


Min:

Max:

Step:

Unit: Seconds

Minor Grid Count Per Unit: 5



Major Grid

Ticks

Is Visible

Color:

Style: Below

Minor Grid

Ticks

Is Visible

Color:

Style: Below

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.

Note that the Legend ItemType setting is ignored for Area charts, they are always colored by Series.

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.

Chart

1 Select Chart Type

2 Select Data

3 Format Chart

Chart Preview

Bar Chart Title

Y-Axis Title

16
12
8
4
0

5.0
4.0
12.0

A
B
C

X-Axis Title

Series 1

Select Chart Type

Bar Chart

Line

Pie

Scatter

Area

Stock

Select Subtype:

Orientation: ☐ Transpose

Multiple Y Axis: ☐ Add Secondary Axis

Dimension: 3D

Angle (0-90):

Output Format: PNG

< Back

Next >

OK

Cancel