Chart Components

Functional Specifications

Draft 1: August 25, 2005

Abstract

This document describes the functional specifications of the Chart Components improvement for the BIRT 2.0 release.

Document Revisions

Draft	Date	Primary Author(s)	Description of Changes
1	08/25/2005	David Michonneau	Initial Draft

Contents

1. Report styles support 3
1.1 Introduction 3
1.2 Use Cases 3
1.3 Style support 3
1.4 Predefined Chart Style 3
1.5 Styles hierarchy 3
1.6 Style properties support 3

2. Labels Overlap 4
2.1 Labels Rotation 4
2.2 Label Staggering 4
2.3 Automatic Label drop 4

3. Legend Title 4
3.1 Feature Description 4
3.2 API Change 5

4. Marker Shapes 5
4.1 Bugzilla Entries 5
4.2 Feature Description 5
4.3 API Change 5

1. Report styles support

1.1 Introduction

BIRT charts have many elements that are similar to elements in other parts of a report. For example, the text used for a chart title, axis title or legend.

To enable the report developer to create consistent styles for all elements of a report, it should be possible to use BIRT styles for elements on a chart.

Use of styles also allows the developer to leverage other benefits -- such as an update of a style will apply to all text that uses that style

1.2 Use Cases

- Create a report. Add a chart and a label. Notice that the label will show its text in "Serif" font, while the chart uses a Sans Serif font. Create a style called "report". Change its font to Cursive. The label changes its font, but the chart does not. Expected to be able to change the report font globally using a style, including charts. Same is true of font size and color, or any style property.
- Create a report with four charts. Decide to set a common font. Common colors. With anything else in BIRT, you can define a style. But, charts don't support styles, so one must manually set each chart and each property separately. Expected charts to support styles since they are an inherent part of BIRT from the user's perspective.

1.3 Style support

The chart item will support one report style, that applies to all of its contents. Note that styles are defined at the BIRT Report level, and therefore only apply when the chart is embedded in a BIRT report. Standalone charts do not support styles.

1.4 Predefined Chart Style

One predefined “chart” style will be defined on the chart item extension. This predefined style will be the default style for all chart items, which do not have any style directly set on them

1.5 Styles hierarchy

The hierarchy is the same as other report items. Here it is as a reminder: the properties are used in the following order, it will go to the next line if the property is undefined at that level.

1- Chart Properties

2- User Style

3- Predefined Chart Style (“chart”)

4- Style of container element of the chart (if any)

5- Predefined Report Style (“report”)

1.6 Style properties support

Not all style properties are supported by charts. Here is a table showing what is supported, and what it applies to inside the chart:

Style Property	Supported	Applies to

Style Property	Supported	Applies to
Font	Yes (all)	All labels
Background	Color and Image only	Chart background
Text Block	No	N/A
Box	Yes	Chart item
Border	Yes	Chart item
Format number	No	N/A
Format Datetime	No	N/A
Format String	No	N/A
Page Break	Before and After only	Chart item
Map	No	N/A
Rules	No	N/A

2. Labels Overlap

X Labels can quickly overlap when too many points are being plotted. This is a very frequent scenario. There are three solutions to that problem:

2.1 Labels Rotation

This is already available. The user can set a rotation angle on the labels to avoid them overlapping.

2.2 Label Staggering

This is typically a manual option set by the user. The labels will appear on two lines alternatively, reducing the incidence of overlap. This only applies if the labels are shown horizontally. The model API already supports this, but the charting engine is not using it yet.

2.3 Automatic Label drop

The last solution when the first two solutions are not used or not sufficient, is to drop some of the X labels (but no points). The engine will drop some X-Axis labels, trying to drop as few as possible. As there is no meaning to have overlapping labels, this will always be active.

3. Legend Title

3.1 Feature Description

https://bugs.eclipse.org/bugs/show_bug.cgi?id=101667

this feature provides the ability to specify a legend title in the legend block section, with its relative position and font attributes.

3.2 API Change

See new schema here: https://bugs.eclipse.org/bugs/attachment.cgi?id=25600

There is a new Title and TitlePosition attribute in the Legend ComplexType:

```
<xsd:complexType name="Legend">
 <xsd:complexContent>
  <xsd:extension base="Block">
   <xsd:sequence>
    ...
    <xsd:element name="Title" type="component:Label" minOccurs="0">
     <xsd:annotation>
      <xsd:documentation>A label instance to hold attributes for legend  title.
       </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
    <xsd:element name="TitlePosition" type="attribute:Position" minOccurs="0">
     <xsd:annotation>
      <xsd:documentation>Specifies where the title for the legend should be  displayed.
       </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
   </xsd:sequence>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

4. Marker Shapes

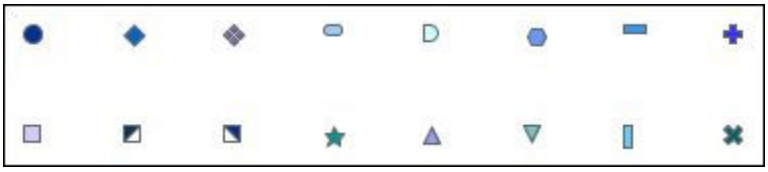
4.1 Bugzilla Entries

https://bugs.eclipse.org/bugs/show_bug.cgi?id=102397

4.2 Feature Description

This will provide a palette of marker icons in addition to the existing marker types (square, circle, triangle, crosshair). The Markers will be able to use that palette of icons to render each serie (there is no “per category” option in that case), so that each point representing a serie will use a given icon from the marker palette.

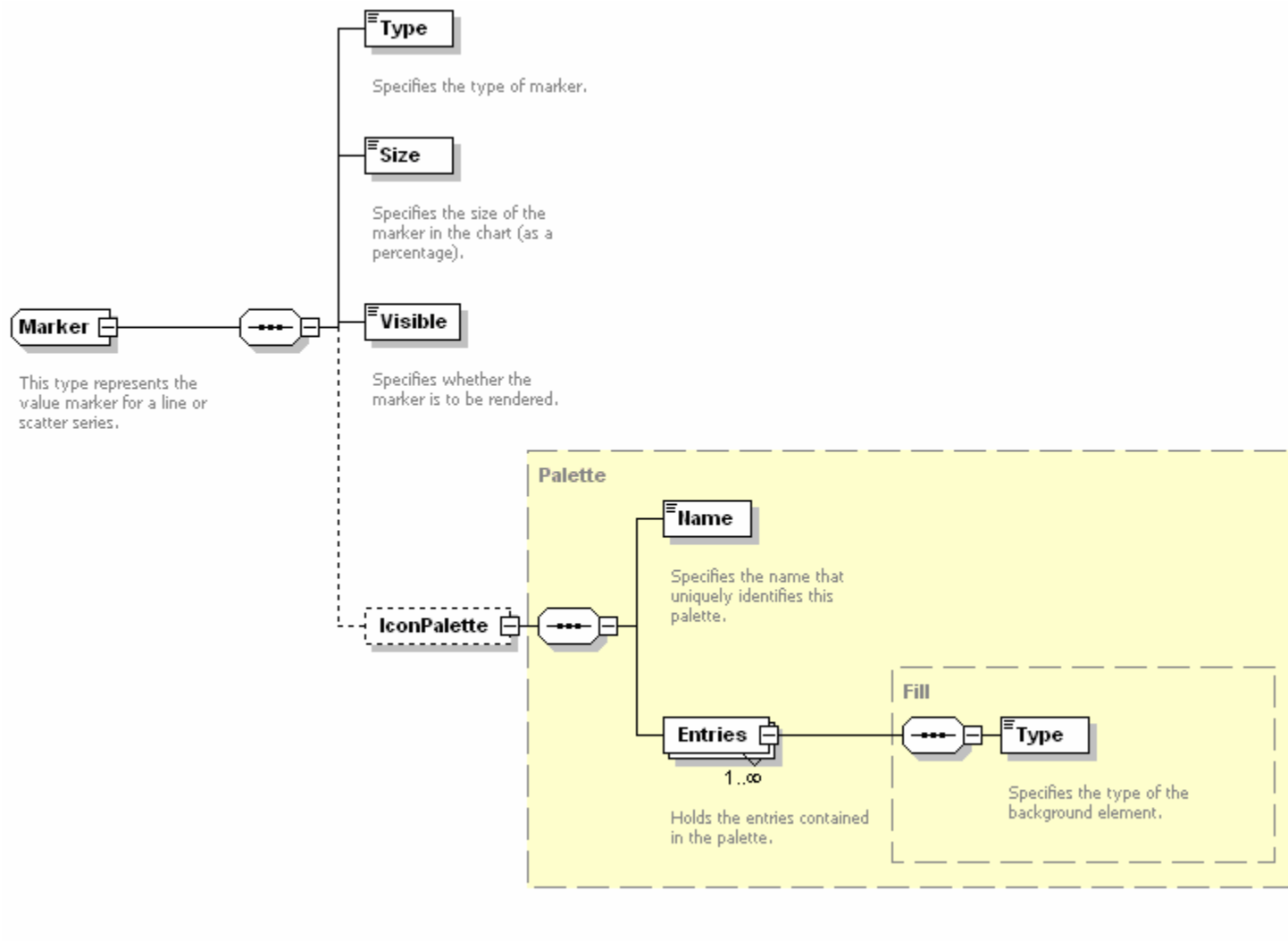
Here are the supported icons that will be provided in the chart builder:



4.3 API Change

Since the Palette already supports images, we only need to associate a separate Palette with the Markers. So that the Markers and the Series can each have their own Palette (in the current model, they share it).

The Marker Type will be extended to support one more type: Icon. In that case, the size will be ignored and the marker will use the icon from the IconPalette, a new Palette attribute on the Marker object. The schema for Marker then becomes as follows:




```
<xsd:simpleType name="MarkerType">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
        This type represents the possible values for markers supported for Line Series.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Crosshair"/>
        <xsd:enumeration value="Triangle"/>
        <xsd:enumeration value="Box"/>
        <xsd:enumeration value="Circle"/>
        <xsd:enumeration value="Icon"/>
    </xsd:restriction>
</xsd:simpleType>


<xsd:complexType name="Marker">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
        This type represents the value marker for a line or scatter series.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
```

```
            <xsd:element name="Type" type="MarkerType">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                    Specifies the type of marker.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="Size" type="xsd:int">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                    Specifies the size of the marker in the chart (as a percentage).
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="Visible" type="xsd:boolean">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                    Specifies whether the marker is to be rendered.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="IconPalette" type="Palette" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
```