BPS 66 - Chart Simple API

Functional Specification

Draft 1: February 13, 2007

Abstract

This document provides an API to easily modify charts inside a report design through the BIRT Report Engine Script API.

Document Revisions

Draft	Date	Primary Author(s)	Description of Changes
1	02/13/2007	David Michonneau	Initial Document

Contents

1. Introduction

1.1 Requirements

The Chart item is built on an extension mechanism and has a complex EMF model, that is independent of the Report model. This makes it difficult for BIRT report users to manipulate chart properties through the API. The first problem is to access the model, which requires some cryptic API calls, and the second issue is to deal with the differences of the Chart and Report Model

A new simple API integrated with the report model script API is therefore required in order to provide a seamless integration for API users. Its goal is to abstract the underlying EMF chart model and provide a more standard interface like for other report items.

1.2 Use Case

This API is intended to be used for modifying existing charts in a report programmatically. Typical steps are as follows:

1- A Report Design is created using the report designer, and contains all needed charts and items

2- In the user application, the report design is opened through the report engine API. An instance of IReportDesign is made available to access the report.

3- The user can access charts easily in the report (and any other extended item), and modify their properties through an easy-to-use simplified model interface, that is consistent with the other report items. For instance the user can modify the data-binding on the chart, the strings and properties of the chart, etc….

1.3 Scope

The assumption is that all the formatting is already done through the Chart Designer, so this API is targeted as changing the contents, not the fonts, colors, margins, etc… This is necessary to keep the API simple enough to use and prevent the user to invalidate the model accidentally.

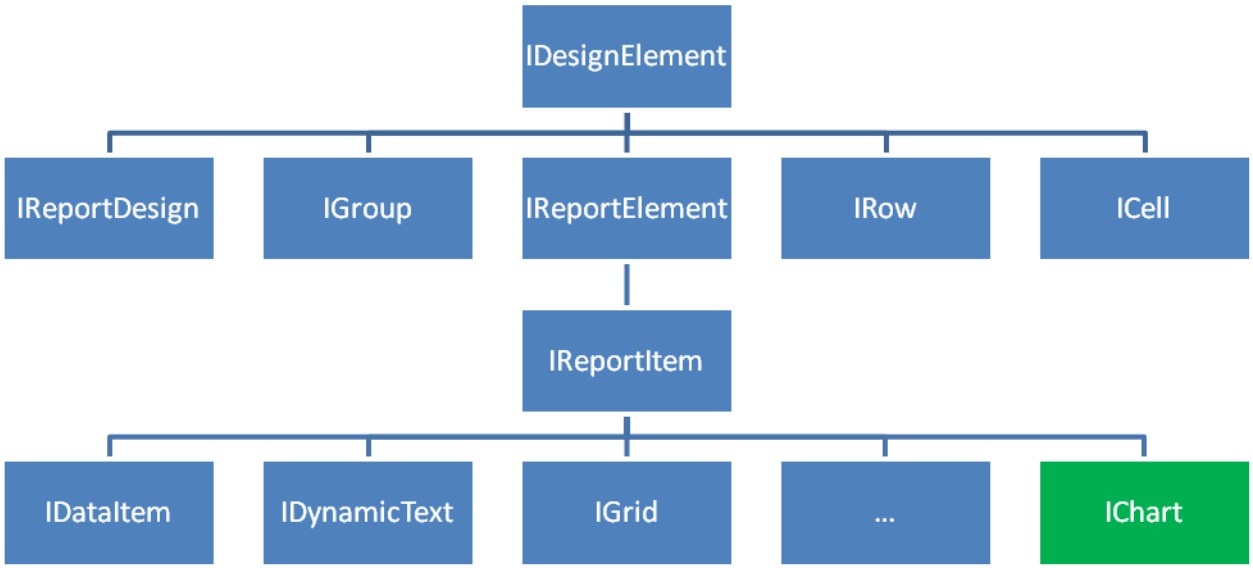
1.4 Extended Items

The framework proposed here can be applied to other extended items, and therefore does not create a hard-coded dependency on the report engine to the charting library, which still plugs in through the OSGi extension framework.

2. Report Engine API Integration

2.1 Class Diagram

A new IChart interface is used to represent the chart model, and fits directly inside the Report Engine Script API model. It is however in a different package, due to its extended nature.



The user can access the chart through **IReportDesign.getReportElement(String name)**

This returns an IReportItem and can then be casted to IChart.

Example:

```
IChart chart = (IChart) rptdesign.getReportElement( "Chart Name" );
```

2.2 Extension Point

A new extension point is needed in org.eclipse.birt.report.engine so that Extended Items can contribute an IReportItem implementation for extended items models.

The description of the extension point is straightforward:

Identifier:

org.eclipse.birt.report.engine.reportitemScript

Since:

2.2

Description:

This extension point allows extended item implementation to provide an interface extending IReportItem to provide API scripting support on the extended item model properties.

Configuration Markup:

```
<!ELEMENT extension (script+)>
<!ATTLIST extension
point CDATA #REQUIRED
id    CDATA #IMPLIED
name  CDATA #IMPLIED>


<!ELEMENT script EMPTY>
<!ATTLIST script
class CDATA #REQUIRED>
```

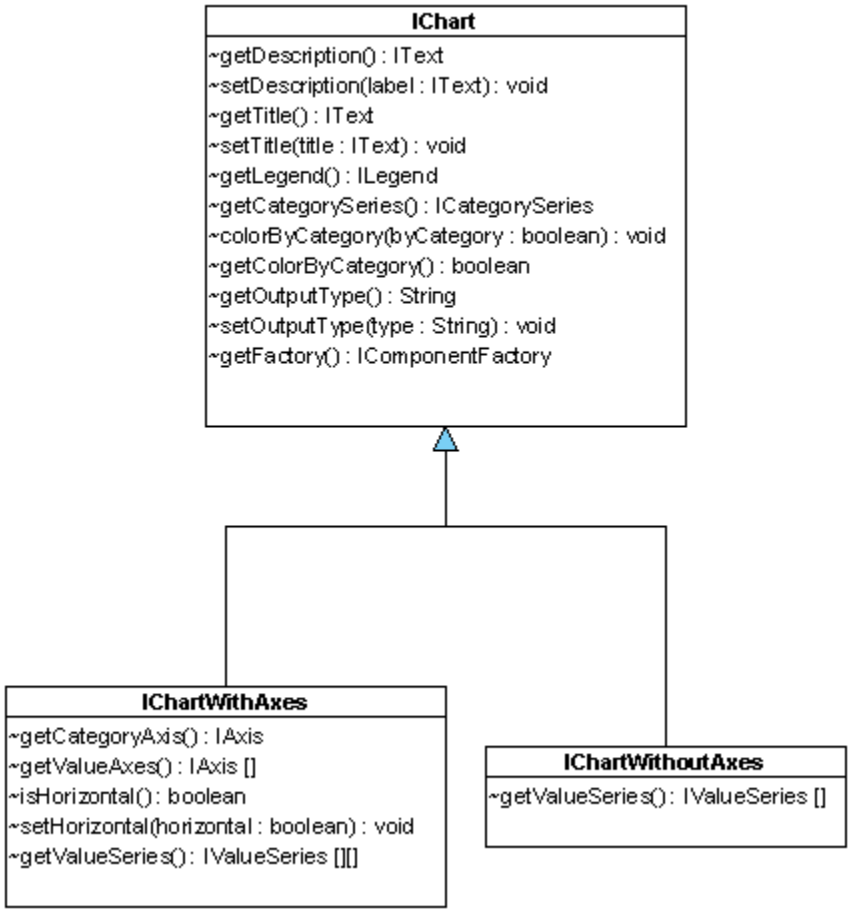
- **class** - The fully qualified name of the java class that implements `org.eclipse.birt.engine.api.script.element.IReportItem`

3. Chart Simple API Reference

This section details the interfaces available in the simple API. There is a strong emphasis on the data related attributes, string values and high-level settings. Formatting setting such as font type, background color, palette color are not available here as they are not likely to change at run-time. Of course conditional formatting (such as changing the color of a point) is to be done through chart scripting and no through this API that has no access to point by point data. A HighlightRule support might be considered in the future.

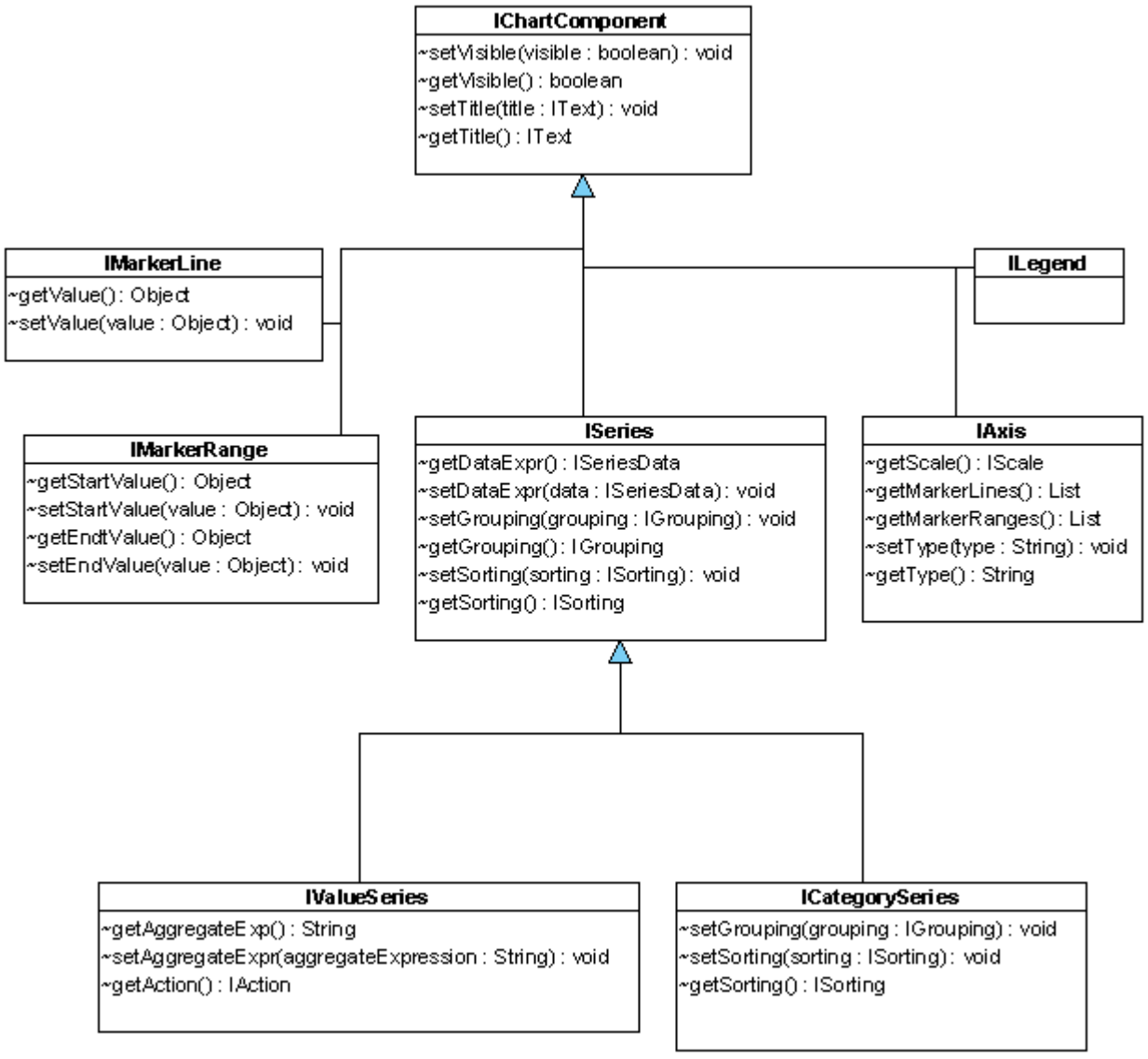
3.1 Main interface

There are two main types of Chart interface, with or without axes. The IChart is the first accessible interface (it directly extends IReportItem).



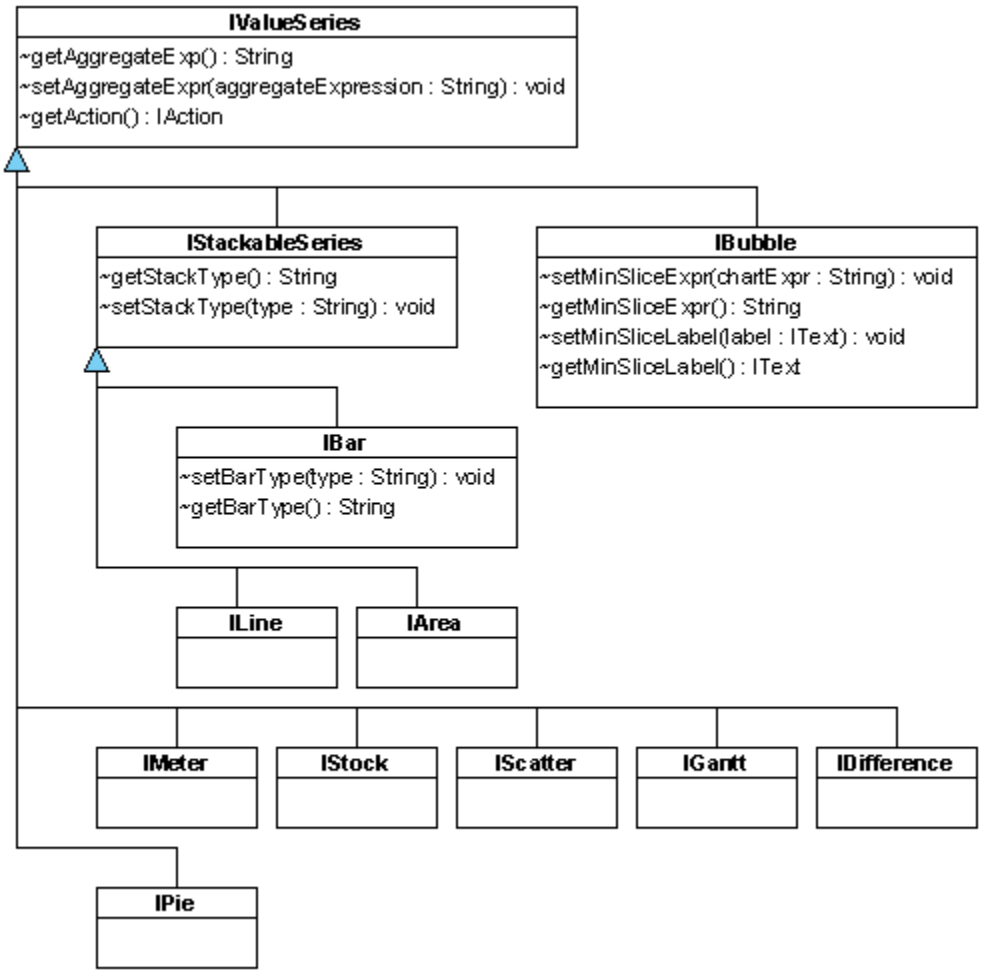
3.2 Components

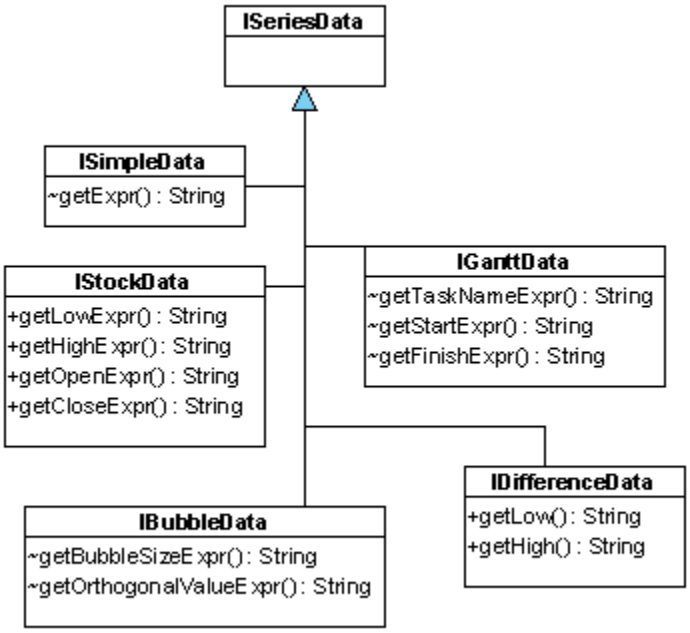
The following diagram shows the various components of the chart model: marker lines, ranges, series, legend, and axes. It's possible to modify the databinding using other expressions and also to access the hyperlink properties through the IAction interface of the engine script api.



3.3 Series

These two diagrams show the various series, and the data they can be bound to. Note that each series uses a specific data binding, for instance stock needs 4 expressions (high, low, open, close).




ISeriesData
ISimpleData
~getExpr() : String
IStockData
+getLowExpr() : String
+getHighExpr() : String
+getOpenExpr() : String
+getCloseExpr() : String
IGanttData
~getTaskNameExpr() : String
~getStartExpr() : String
~getFinishExpr() : String
IBubbleData
~getBubbleSizeExpr() : String
~getOrthogonalValueExpr() : String
IDifferenceData
+getLow() : String
+getHigh() : String

3.4 Scale

This diagram shows the various possible scales: time scale, linear, logarithmic or category. The scales can be automatic or be manually set.