BPS 43 - Chart Web Components

Functional Specifications

Draft 1: October 24, 2006

Abstract

This project proposes new Web components aimed at integrating charts into web applications from both a client and server perspectives: on the client side, an AJAX-based library to embed charts in HTML pages. On the server side, a JSP tag library will make it easy to integrate chart rendering in a Java web application.

Document Revisions

Draft	Date	Primary Author(s)	Description of Changes
1	10/24/2006	David Michonneau	Initial Draft

Contents

1. Introduction 3

2. Chart AJAX API 3
2.1 Environment 3
2.2 Integration in Web Pages 3
2.3 Live Feature 4
2.4 Back-end Configuration 4
2.5 Chart AJAX API reference 4
2.5.1 class Chart 4
2.5.2 class Connection 5

3. JSP Tag library 6
3.1 JSP Chart Tag 6
3.1.1 Required Attributes/SubTags 6
3.1.2 Optional attributes/Subtags 6
3.1.3 Servlet/Image Caching 7
3.1.4 Configuration 7
3.1.5 Example 7

1. Introduction

This project proposes two chart libraries:

- An AJAX based library as a javascript file.
- A JSP tag library.

Both libraries aim at integrating charts easily into web pages but with different technologies and architecture. The AJAX one is targeted to generic HTML pages, can be controlled through standard JavaScript code and works with charts inside BIRT reports. The JSP tag library is a server side library to be used within JSP pages and work with standalone chart models.

2. Chart AJAX API

The Chart AJAX API lets you embed interactive charts in your HTML pages.

2.1 Environment

This feature requires a BIRT report engine on the back-end. The supported browsers are IE6, IE7, Firefox 1.5 and 2.0. The HTML pages can be hosted on any web server. the only requirement is that the report engine is on the same web server where the javascript file is hosted.

2.2 Integration in Web Pages

To use the Chart AJAX API in your web page, you simply need to access the chartapi.js file in the HTML page and use the available JavaScript API functions together with div elements.

Here is an example:

```
<html>
  <head>
    <title>Chart Web API Example</title>
    <script src="http://myserver/js/chartapi.js"
            type="text/javascript"/>
    <script type="text/javascript">

    function load() {
    {
           var connection = new Connection(http://myserver);
           var chart = new
    Chart(document.getElementById("chart"));
           chart.setConnection( connection );
           chart.setSource( "/myrptdesign.rptdesign",
"chartreportlet" );

    }

    </script>
  </head>
  <body onload="load()">
    <div id="chart"></div>
  </body>
</html>
```

It will display the corresponding chart in the place of the div element “chart”. Of course it’s possible to display any number of charts within the same page.

2.3 Live Feature

In addition to the interactive features of the BIRT charts, the charts rendered through the AJAX API optionally support live refresh.

The chart can be refreshed automatically by setting a time interval. This is only enabled if the source of the chart is a report design file, and not a report document.

See `setRefreshInterval( Integer seconds )` in the Chart AJAX API reference section.

2.4 Back-end Configuration

The charts must be contained within a BIRT report design or report document. These reports are hosted on the same Web application server as the BIRT Report Engine.

The main requirement is that the chart must be referenced by a bookmark in the report so that they can be accessed as a reportlet. That is how the Chart AJAX API extracts the charts and displays them on the web page.

2.5 Chart AJAX API reference

The Chart AJAX API uses an Object-Oriented design for ease of use. Here are the classes and methods available.

2.5.1 class Chart

Constructor	**Description**
`Chart( container )`	Creates a new Chart in an HTML container, usually a div element.

Methods	**Description**
`setConnection( Connection connection)`	Set the connection object used to access the iServer host.
`setSource( String reportName, String reportlet )`	Specifies the path of the report and the value of the reportlet used to reference the chart
`setRefreshInterval( Integer seconds )`	For live charts, this will automatically refresh the chart every n seconds. If set to 0 (default), it will never refresh it

2.5.2 class Connection

Constructor	Description
`Connection( String serverURL )`	Creates a new connection to BIRT Report Engine at the specified URL.

3. JSP Tag library

3.1 JSP Chart Tag

A JSP tag library is provided with a chart tag in order to display a chart on a web page. Several tag attributes and subtags are to be specified to configure the output:

3.1.1 Required Attributes/SubTags

Name	Attribute/Subtag	Type	Description
Model	Tag	String or Chart	The chart model as XML (relative path) or java instance
width	Attribute	double	Width in points
height	Attribute	double	Height in points
renderURL	Attribute	String	Path to the RenderChart servlet, as specified in the servlet config.

3.1.2 Optional attributes/Subtags

Name	Attribute/ Subtag	Type	Description	Default
Data	Tag	IDataRowExpression or java.sql.ResultSet	Data to bind to chart model.	Runtime Data in chart model instance
output	Attribute	String	The output type (SVG, PNG, BMP, JPG, PDF).	PNG
ExternalContext	Tag	IExternalContext	Context to be used in chart scripting.	implementation wrapping the javax.servlet.jsp. PageContext
StyleProcessor	Tag	IStyleProcessor	Used to enable styling of the chart	None
RuntimeContext	Tag	RunTimeContext	Used to set various runtime attributes for the chart generation and rendering	None

3.1.3 Servlet/Image Caching

The tag itself only renders the html tag referencing the image that is still on the server side. The image (static image, SVG or PDF) is then streamed back by a servlet, that is managing the storage of the image as well.

The image can be stored on disk or in session, depending on the servlet configuration. A timeout is also indicated for the time to keep the image available to the browser after it has been read once. When the session or timeout expires, the image is cleaned up from session or disk.

3.1.4 Configuration

```
<servlet>
    <servlet-name>RenderChart</servlet-name>
    <servlet-class>org.eclipse.birt.chart.servlet.ChartRenderer</servlet-
class>
    <!—- Caching model : session or disk -->
     <init-param>
        <param-name>caching</param-name>
        <param-value>session</param-value>
    </init-param>
    <!—- Caching timeout in minutes (applies after the image is read once)
-->
     <init-param>
        <param-name>timeout</param-name>
        <param-value>0</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
    <servlet-name>RenderChart</servlet-name>
    <url-pattern>/chart/*</url-pattern>
</servlet-mapping>
```

3.1.5 Example

```
<%@ taglib uri="BIRTChart" prefix="c" %>
…
<c:chart width="100" height="50" output="PNG" renderURL="chart">
        <c:model>/chartmodels/chartexample.chart</c:model>
        <c:data><%= dataFeed %></c:data>
</c:chart>
```