

# BIRT Chart Interactivity

Functional Specifications

Draft 1: July 21, 2005

## Abstract

*This document describes the functional specifications of the Chart interactivity features for BIRT 2.0*

## Document Revisions

Draft	Date	Primary Author(s)	Description of Changes
1	July 21, 2005	David Michonneau	Initial Draft

## Contents

<b>1. Introduction.....</b>	<b>3</b>
<b>2. Use Cases.....</b>	<b>3</b>
2.1 Highlight .....	3
2.2 Visibility Toggle .....	3
2.3 URL redirection .....	3
2.4 ToolTip .....	3
<b>3. Triggers .....</b>	<b>3</b>
<b>4. Actions.....</b>	<b>4</b>
4.1 Description .....	4
4.1.1 Show ToolTip.....	4
4.1.2 Url Redirect .....	4
4.1.3 Toggle Visibility.....	4
4.1.4 Invoke Script.....	4
4.1.5 Highlight (new).....	4
4.1.6 Zoom (new) .....	4
4.1.7 3D Rotation (new) .....	4
4.2 Support matrix.....	5
<b>5. Scripting .....</b>	<b>5</b>
5.1 SVG Scripting.....	5
5.1.1 SVG JavaScript API .....	5
5.2 Static Images Scripting .....	6
5.2.1 Static images Script API.....	6
<b>6. Chart Interactivity API .....</b>	<b>6</b>
6.1 InteractionEvent .....	6
6.2 Trigger.....	7
6.3 Action .....	7
<b>7. BIRT HTML Viewer Interactivity .....</b>	<b>8</b>
7.1 Browser Requirements .....	8
7.2 Automatic SVG support detection .....	8
<b>8. Built-in Interactivity .....</b>	<b>8</b>
8.1 Top Menu .....	9
8.2 Chart Types.....	9
8.3 3D Rotation .....	9
8.4 Zoom .....	9
8.5 Visibility .....	9
8.5.1 Hide serie on selection .....	10
8.5.2 Highlight serie on selection.....	10
<b>9. Chart Builder .....</b>	<b>10</b>

## 1. Introduction

The chart interactivity allows the users to perform different actions on the chart they are viewing.

Each action is mapped to a user gesture that we call a trigger. Triggers can be mouse or keyboard events. The Actions can modify the visuals of the chart in different ways, but they can also perform some change at the viewer level, such as redirect to a new page for instance.

Interactivity features depend on the output type, but also on the environment (report/html/birt viewer). The goal is to achieve the best interactivity using SVG inside the HTML BIRT viewer. Other outputs will benefit some interactivity features as well.

## 2. Use Cases

These use cases are extracted from the Chart interactivity design document of IBM.

### 2.1 Highlight

A pie chart is displayed in the eclipse workbench. Therefore the chart is rendered in an SWT device. The user has the ability to select a pie slice. When the user select a pie slice the pie slice changes color indicating that it has been selected. Furthermore, property information associated with the pie slice is displayed in the property view.

### 2.2 Visibility Toggle

A line chart consisting of two datasets is displayed in an html page using the SVG renderer. The user has the ability to select a legend item to toggle the visibility of the associated dataset in the plot area.

### 2.3 URL redirection

A pie chart is displayed in an html page using the SVG renderer. When the user selects a pie section the browser is redirected to a URL.

### 2.4 ToolTip

A pie chart is displayed in the eclipse workbench. When the user hovers over a pie section tooltip text is displayed.

## 3. Triggers

Here are the possible triggers:

- Mouse click

- Mouse hover
- Mouse double-click (new)
- Key down/up (new)
- Script trigger (new): this trigger allows any kind of event that the script environment can support. It will of course depend on which renderer is used and the capabilities of the script engine.

Each trigger is associated with any visual component of the chart, called hotspot: that can be a legend item, the chart title, a chart serie, etc... Each hotspot will react to some predefined triggers with one or several actions. Note that this action is not limited to the hotspot it originated.

## 4. Actions

Actions are executed in answer to some trigger events.

### 4.1 Description

Here are the possible actions:

#### 4.1.1 Show ToolTip

This shows a tooltip on a chart element. It is normally associated with a mouse hover event.

#### 4.1.2 Url Redirect

Only works in an html environment. This redirects the browser to a new url. A possible extension is to redirect the browser to a different page of the report or another report.

#### 4.1.3 Toggle Visibility

This allows to toggle the visibility of one or several series

#### 4.1.4 Invoke Script

This invokes a script inside the viewer. The scripts are written in a language specific to the rendering environment and output type.

#### 4.1.5 Highlight (new)

This highlights a chart element. It is normally associated with a mouse click event.

#### 4.1.6 Zoom (new)

This allows the user to zoom in and out. It is particularly useful for complex charts. The Zoom is done on the whole chart, but can be centered on a particular element.

#### 4.1.7 3D Rotation (new)

This applies to 3D charts and allows a 3D rotation

## 4.2 Support matrix

Not all actions are supported for all outputs and environments. Here is a table summarizing the actions on an output basis.

Output Type/Action	SVG	SWT	Swing	Static image in PDF	Static image in HTML
Url Redirect	✓	✓	✓	✓	✓
Show Tool tip	✓	✓	✓	✓	✓
Toggle Visibility	✓	✓	✓		
Highlight	✓	✓	✓		
Zoom	✓	✓	✓		
3D Rotation	✓	✓	✓		
Invoke Script	✓				✓

## 5. Scripting

When the built-in interactivity features defined by the API are not enough, the user can write his own interactivity script, which is dependent on the output type. This is not to be confused with Chart Scripting which is done at the rendering time.

This release will provide scripting support for SVG (in any svg viewer) and static images (inside a browser)

### 5.1 SVG Scripting

SVG output supports the ECMAScript language, also known as JavaScript.

An API will be provided to allow the user to manipulate the SVG DOM and access the chart engine on the server.

The capabilities of SVG Scripts allow it to communicate with the chart engine at a specific url through web services. It can also modify the SVG output

#### 5.1.1 SVG JavaScript API

TBD

## 5.2 Static Images Scripting

It's possible to write scripts for static images, using JavaScript. An API will be provided to communicate with the chart engine or access other report elements. These scripts can only be used in a javascript-enabled environment.

### 5.2.1 Static images Script API

TBD

## 6. Chart Interactivity API

Several classes in the Chart API define the interactivity properties. These properties are then used by each device renderer. Here are the key classes and interfaces to define interactivity at the API level:

### 6.1 InteractionEvent

The InteractionEvent defines the triggers and action mapping on given hotspots:

public final class **InteractionEvent**

extends [PrimitiveRenderEvent](#)

**Author:**

Actuate Corporation

**See Also:**

[Serialized Form](#)

### Field Summary

Fields inherited from class org.eclipse.birt.chart.event.[PrimitiveRenderEvent](#)

[DRAW](#), [FILL](#), [iObjIndex](#)

### Constructor Summary

[InteractionEvent](#)(java.lang.Object source)

### Method Summary

void	<a href="#">addTrigger</a> ( <a href="#">Trigger</a> t)
<a href="#">Action</a>	<a href="#">getAction</a> ( <a href="#">TriggerCondition</a> tc)

<a href="#">PrimitiveRenderEvent</a>	<a href="#">getHotSpot</a> ()
<a href="#">Trigger</a> []	<a href="#">getTriggers</a> ()
void	<a href="#">reuse</a> (java.lang.Object oNewSource)
void	<a href="#">setHotSpot</a> ( <a href="#">PrimitiveRenderEvent</a> pre)

## 6.2 Trigger

public interface **Trigger**

extends org.eclipse.emf.ecore.EObject

A representation of the model object '**Trigger**'. This type defines a Trigger. A trigger defines interactivity for a chart component.

The following features are supported:

- [Condition](#)
- [Action](#)

See Also:

[DataPackage.getTrigger\(\)](#)

### Method Summary

<a href="#">Action</a>	<a href="#">getAction</a> () Returns the value of the ' <b>Action</b> ' containment reference.
<a href="#">TriggerCondition</a>	<a href="#">getCondition</a> () Returns the value of the ' <b>Condition</b> ' attribute.
boolean	<a href="#">isSetCondition</a> () Returns whether the value of the ' <a href="#">Condition</a> ' attribute is set.
void	<a href="#">setAction</a> ( <a href="#">Action</a> value) Sets the value of the ' <a href="#">Action</a> ' containment reference.
void	<a href="#">setCondition</a> ( <a href="#">TriggerCondition</a> value) Sets the value of the ' <a href="#">Condition</a> ' attribute.
void	<a href="#">unsetCondition</a> () Unsets the value of the ' <a href="#">Condition</a> ' attribute.

## 6.3 Action

public interface **Action**

extends org.eclipse.emf.ecore.EObject

A representation of the model object '*Action*'. This type defines an Action. An action is a property defining interactivity for an element. It is associated in a trigger with a trigger condition that defines when the action is to be processed.

The following features are supported:

- [Type](#)
- [Value](#)

See Also:

[DataPackage.getAction\(\)](#)

## Method Summary

<a href="#">ActionType</a>	<a href="#">getType</a> () Returns the value of the ' <i>Type</i> ' attribute.
<a href="#">ActionValue</a>	<a href="#">getValue</a> () Returns the value of the ' <i>Value</i> ' containment reference.
boolean	<a href="#">isSetType</a> () Returns whether the value of the ' <i>Type</i> ' attribute is set.
void	<a href="#">setType</a> ( <a href="#">ActionType</a> value) Sets the value of the ' <i>Type</i> ' attribute.
void	<a href="#">setValue</a> ( <a href="#">ActionValue</a> value) Sets the value of the ' <i>Value</i> ' containment reference.
void	<a href="#">unsetType</a> () Unsets the value of the ' <i>Type</i> ' attribute.

## 7. BIRT HTML Viewer Interactivity

### 7.1 Browser Requirements

Full interactivity will only be available when the user is using a SVG and JavaScript enabled browser. The Chart full interactivity will be available on IE6+ with Adobe SVG Plugin or Firefox 1.1.

### 7.2 Automatic SVG support detection

The report will automatically show SVG content if the browser supports it, otherwise it will use a static image of the chart, with limited interactivity.

## 8. Built-in Interactivity

BIRT will provide default built-in interactivity for the Chart, in addition to what the user defines. This section describes what is this default interactivity.



## 8.1 Top Menu

A menu bar will be displayed over the chart to allow several actions on the chart. Here is a short tree describing the Menu structure:

- Chart Type
  - Bar
    - Side by Side
    - Stack
    - Stack Percent
  - Line
    - Side by Side
    - Stack
    - Stack Percent
  - Area
  - Pie
  - Stock
  - Scatter
- Zoom
  - Zoom in
  - Zoom out
- Visibility
  - Hide serie on selection (default)
  - Highlight serie on selection

## 8.2 Chart Types

The top menu will allow to change the type of the chart. It will ask the chart engine to regenerate it.

## 8.3 3D Rotation

3D Charts can be rotated on any 3D axis by clicking on any part of the chart and holding the mouse button while moving the mouse.

## 8.4 Zoom

The zoom can be accessed through the top menu

## 8.5 Visibility

There are exclusive choices

**8.5.1 Hide serie on selection**

Clicking on a legend item or serie will hide/show the corresponding serie

**8.5.2 Highlight serie on selection**

Clicking on a legend item or serie will highlight the corresponding serie.

**9. Chart Builder**

The Chart builder will have an additional tab for interactivity settings.

TBD