**Jsf4Birt components user guide**

**Birt wrapper component**

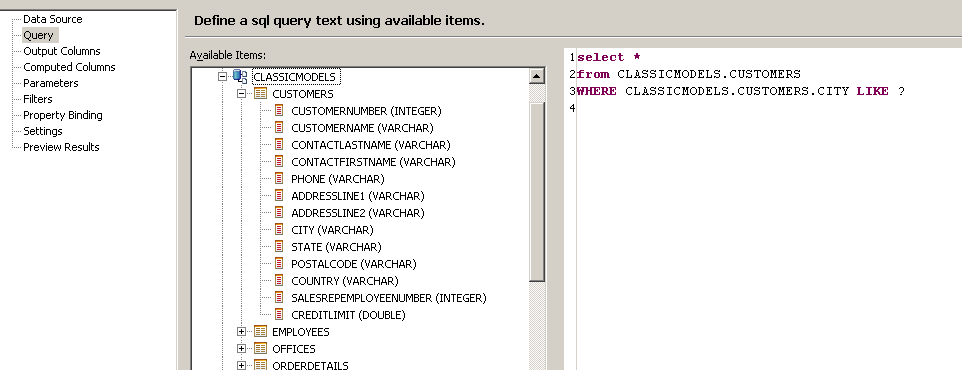
Birt wrapper designed to display BIRT reports created in BIRT eclipse designer. This component uses BIRT runtime API which is required as preparation before component usage(see Deploying JSF Actuate library in J2EE application.docx document for details). Birt wrapper component calls appropriate API from BIRT runtime and gets generated HTML code and puts it on the page. This job fully is done on server side.

Supported version of BIRT runtime: 2.3.\* & 2.5.

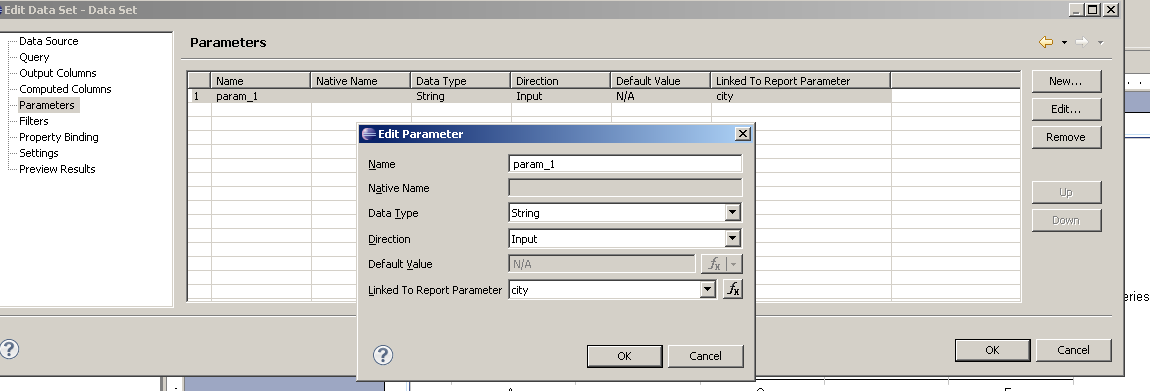
Usage example

BIRT runtime contains DB sample which contains some classic schemas. Let’s take a look at CUSTOMER table. It includes customer name (first/last), city, credit limit & etc. Imagine that we would like to create BAR chart displaying credit limit for all customers from city defined from application user interface. So, firstly we should create report in eclipse designer:

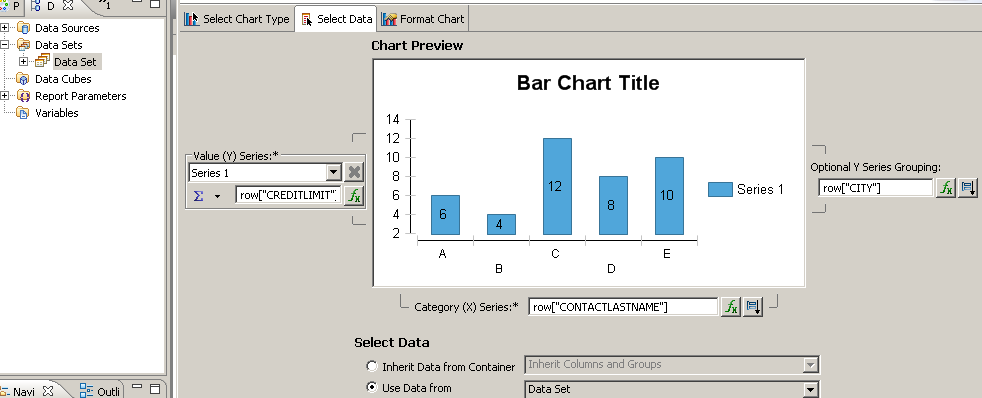
1. Define data set



DB query contains one ‘city’ parameter. We should bind it to report parameter



1. Add BAR chart to report add bind data



**Ready report design file here: **

After report design is completed we have to put report under server. (Tomcat 6 for example). BIRT runtime & jsf4Birt library must be installed according to Deploying JSF Actuate library in J2EE application.docx document.

Now we can use jsf4Birt tag library to display report in our page.

<h:form>

<h:inputText value=*"#{bean.city}"* /><br/>

<h:commandLink value=*"Search"* />

<birt:birtWrapper reportDesign=*"/Reports/new\_report\_8.rptdesign"*>

<f:param name=*"city"* value=*"#{bean.cityParam}"*/>

</birt:birtWrapper>

</h:form>

<birt:birtWrapper> tag defines placeholder on the page where generated report will be rendered.

‘reportDesign’ attribute specifies relative path to report design file.

‘city’ parameter should be passed to report through <f:param> tag.

Bean class code:

public class Bean {

String city;

**public** String getCity() {

**return** city;

}

// we should return parameter with ‘%’ suffix for SQL LIKE operator //syntax

**public** String getCityParam () {

**return** city != **null** ? city + "%" : **null**;

}

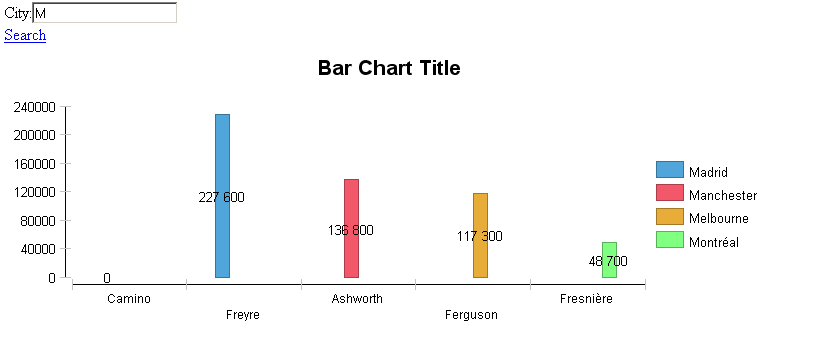
**public** **void** setCity(String city) {

**this**.city = city;

}

}

After application run we can enter first initial for city and press ‘Search’ button. Report will be updated and will display chard according to parameter entered:



We can specify CSS styles, width, height & etc for birt wrapper tag. See tlddoc for details.

**Actuate wrapper component**

Actuate wrapper designed for loading reports from Actuate server installed. Component uses Actuate JS API available via link (<http://actuate.server.url/jsapi>) to load and run reports. Detailed documentation for actuate JS API you can find here: <http://www.birt-exchange.org/documentation/JavaComponents10/JSAPI-JSdoc/>

Usage example:

To use actuate wrapper we need Actuate server installed. Jsf4Birt component library should be added to application classpath.

NOTE: BIRT runtime is not required to use actuate wrapper component.

Page code:

<birt:actuateWrapper id=*"id"*

width=*"900px"*

reportName=*"#{bean.reportName}"*

height=*"700px"*

style=*"float: left; position: relative"*>

<f:facet name=*"emptyReport"*>

<h:panelGroup layout=*"block"* style=*"background-color: green"*>

<table style=*"width: 100%; height: 100%"*>

<tr>

<td>Select a report</td>

</tr>

</table>

</h:panelGroup>

</f:facet>

<birt:actuateUIOption name=”*enabledMainMenu”* value=*”false”* />

<f:facet name=*"header"*>

<h:panelGroup layout=*"block"* >

<h:outputText value=*”Component Header”* />

</h:panelGroup>

</f:facet>

</birt:actuateWrapper>

‘reportName’ attribute defines path to report file on actuate server

‘emptyReport’ facet defines area will be rendered if ‘reportName’ is null. In current example we display ‘Select a report’ message in the center of the report container.

‘header’ facet defines component header will be rendered above the report container.

<birt:actuateUOption> switches off the main menu for the report. This tag can specify any UI option available via actuate JS API. See here: <http://www.birt-exchange.org/documentation/JavaComponents10/JSAPI-JSdoc/symbols/actuate.viewer.UIOptions.html> for details.

URL, login and password for actuate server can be configured though:

1. context parameters in web.xml or

<!-- Birt settings -->

<context-param>

<param-name>actuate.serverUrl</param-name>

<param-value>http://localhost:8900/iportal</param-value>

</context-param>

<context-param>

<param-name>actuate.serverLogin</param-name>

<param-value>administrator</param-value>

</context-param>

<context-param>

<param-name>actuate.serverPassword</param-name>

<param-value></param-value>

</context-param>

1. definition of component attributes (see tlddoc for details)

Component will load actuate JS API from actuate server, initialize it and run report specified:

