

Dashboard 构建指南

原文版本 **1.2RC3**

原文链接 [网址](#)

翻译 王娜

校对 覃辉

Pentaho 中文讨论组

QQ 群: 12635055, 18806621

论坛: <http://www.bipub.org/bipub/index.asp>

1 介绍

这篇文档描述了如何使用 Pentaho BI 套件创建 dashboards。它描述了 Pentaho Demo 服务器提供的 sample dashboard。

sample dashboard 以 JSP 和 PHP 形式提供。两个 dashboards 均使用相同的内容，提供相同的功能。JSP 实例显示了如何在 Pentaho UI 组件中使用 Java API，PHP 实例显示了如何在 Pentaho UI 组件中使用 Web 服务。

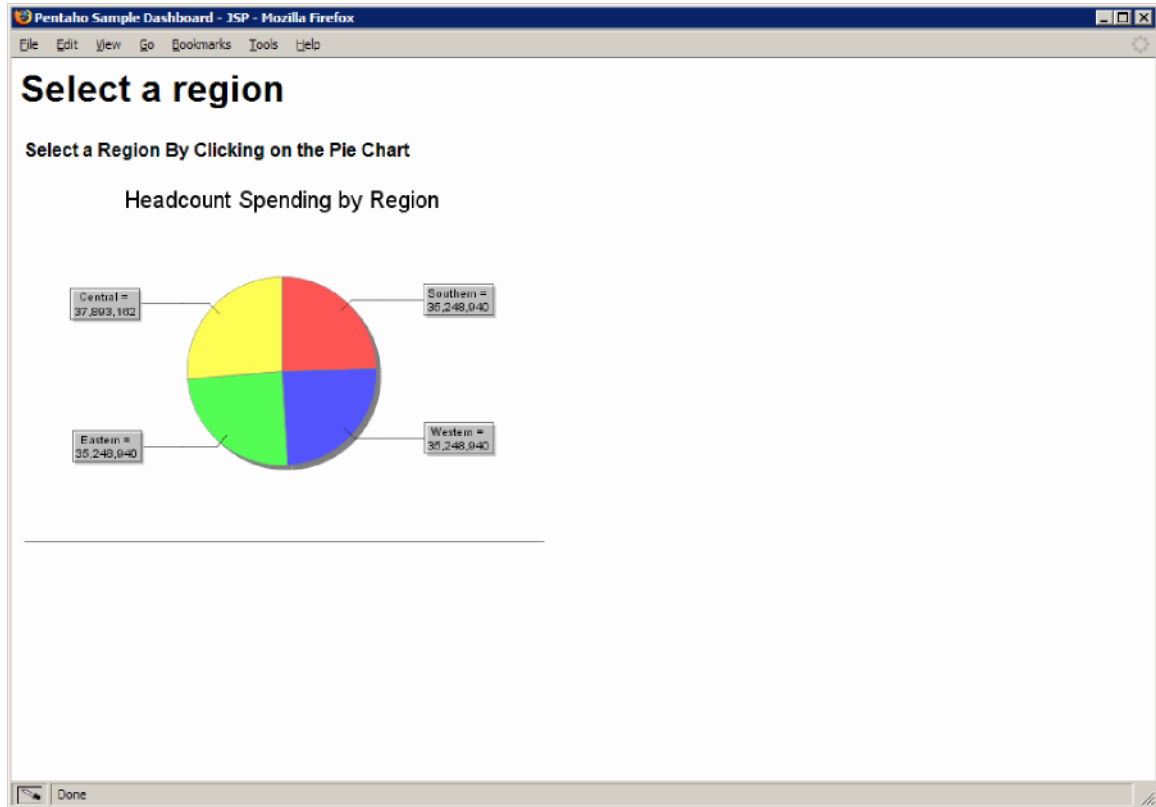
Java API 实例可用于 JSPs, Servlets, 或 Java 应用。Web 服务实例可用于可发布 web 服务调用的任何技术，例如 PHP, IBM Domino 等。

2 Sample Dashboard

Sample Dashboard 位于这个 URL (假设你将 Pentaho demo 安装于你本机的 8080 端口)

<http://localhost:8080/pentaho/jsp/SampleDashboard.jsp>

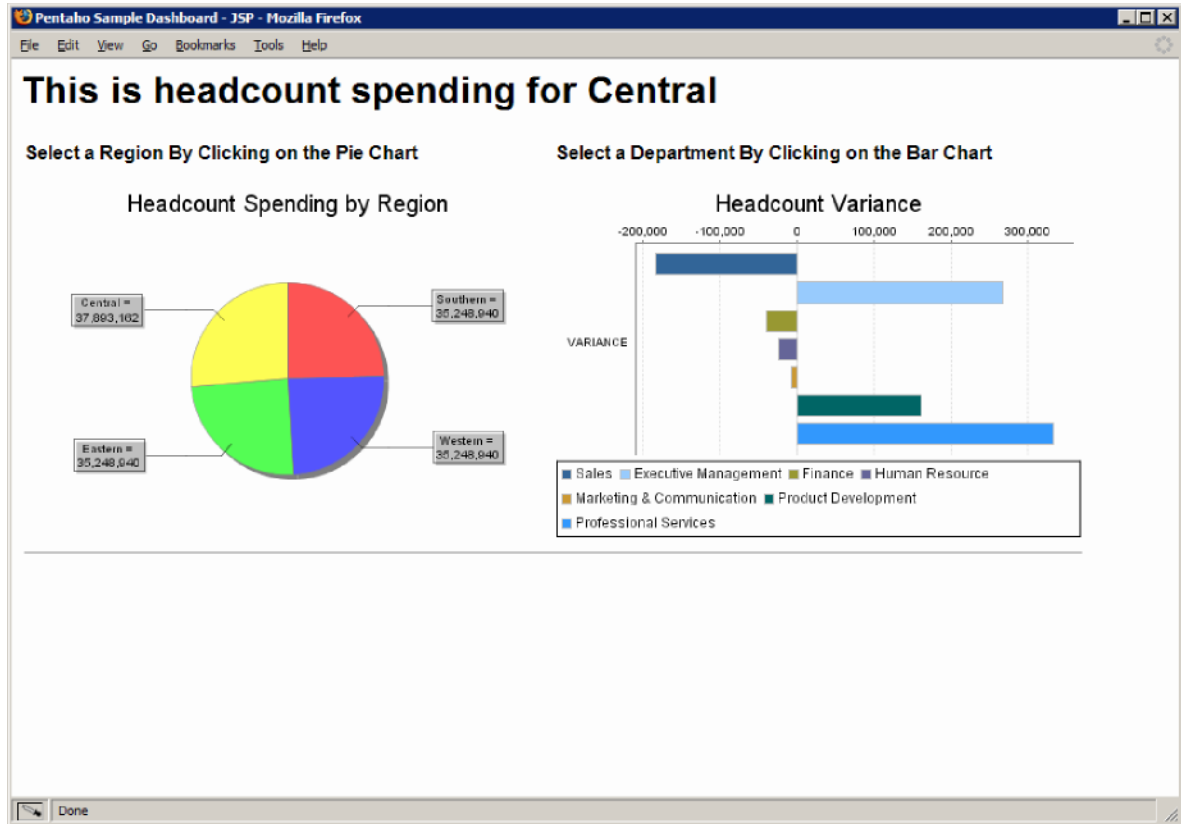
sample dashboard 首先显示了一张饼图，其显示了四个区域中每个的 headcount costs。



关于如何定义饼图的更多信息请参考下面的 'Headcount Spending by Region Pie Chart'。

关于如何获得饼图的数据的更多信息请参考下面的 'Headcount Spending by Region Action Sequence'。

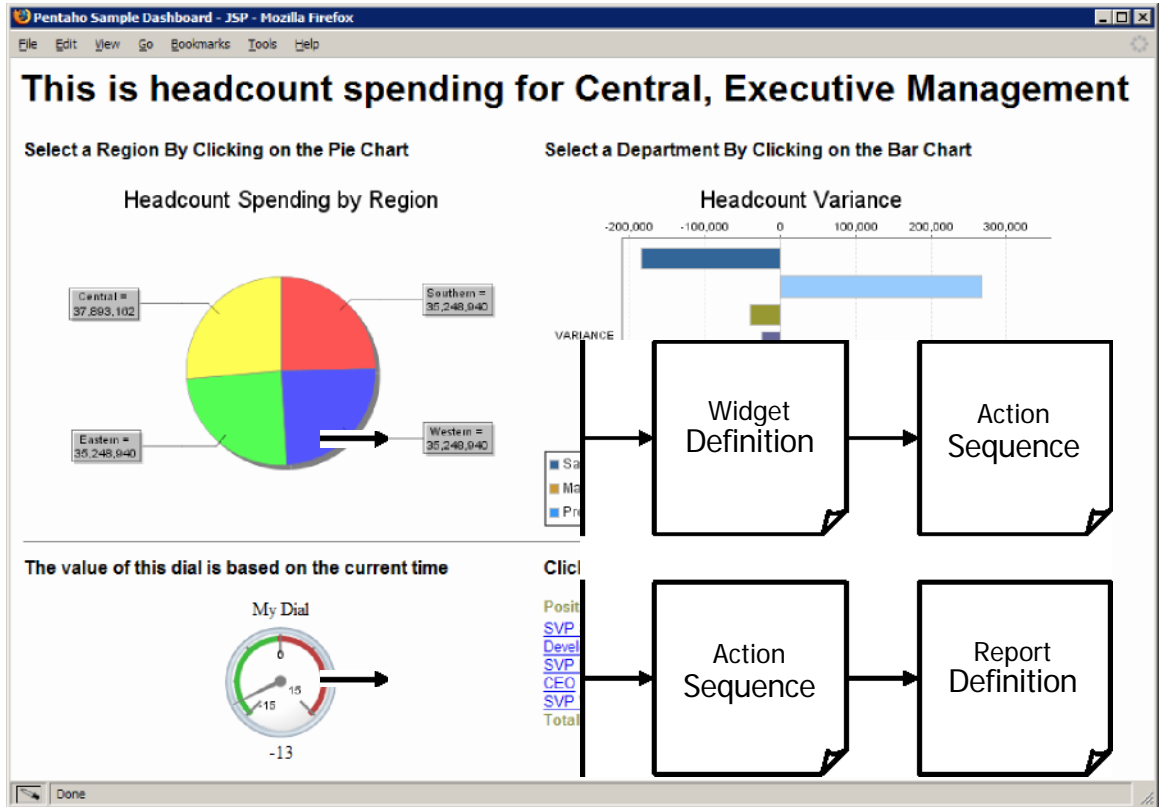
当用户点击饼图上的一个切片时，出现了另一张饼图，对于选中的区域的每个部门，显示了 headcount budget 和 actual cost 间的不同。



饼图如何被定义的更多信息请参考下面的 'Headcount Variance Pie Chart'。

如何获取饼图的数据的更多信息请参考下面的 'Headcount Variance Action Sequence'。

当用户点击一个部门的 bar 时，dashboard 在图表下，显示了一个 dial 和数据的一个表。dial 基于当前时间(因此它随着时间改变)显示了一个值，表显示了选中部门内的 positions 的详细信息。



如何定义 dial 的更多信息请参考下面的 'Dynamic Dial'。

如何定义嵌入式报表的更多信息请参考下面的 'Embedded Report'。

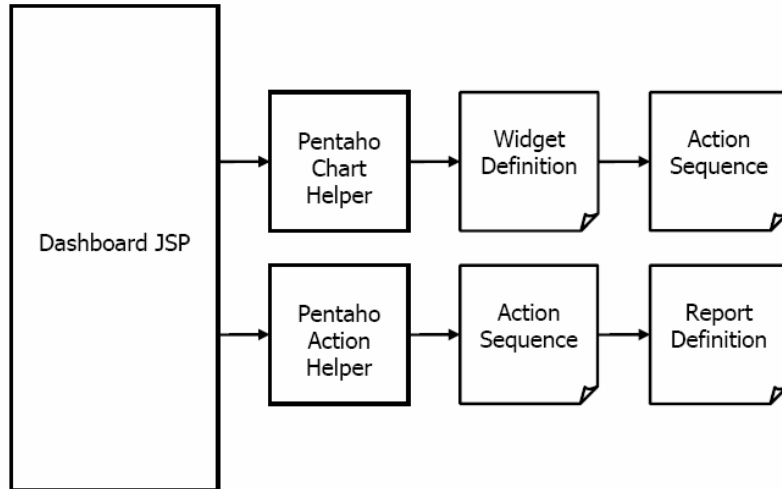
当用户点击表中的一个 position 时, dashboard drills 到另一个页面, 并传送 region, department 和 position 作为参数。



这一页的源代码请参考附录 8: Drill JSP

3 体系结构

这个 sample dashboard 是一个 JSP，其使用 Pentaho API 组件以生成这页上的图像和表。



4 内容定义

4.1 Headcount Spending By Region Pie Chart

这张饼图的定义在 `pentaho-demo/pentaho-solutions/samples/dashboard/regions.widget.xml`

这个 XML 文件定义了：

设置	描述	值
chart/title	图的标题	Headcount Spending by Region
chart/title-position	图的标题的位置(上, 下, 左或右)	top
chart/title-font/font-family	标题字体的名字	Ariel
chart/title-font/size	标题字体的大小	20
chart/title-font/is-bold	标题字体是粗体么?	false
chart/title-font/is-italic	标题字体是斜体么?	false
chart/is-3D	饼图是 3D 的么?	false
chart/border-visible	Does chart have a border?	false
chart/include-legend	Does chart have a legend	false
chart/data/data-solution	action sequence 的 solution 为这个图提供了数据	samples
chart/data/data-path	action sequence 的路径	dashboard
chart/data/data-action	action sequence 的名字	regions_headcount_data.xaction
chart/data/data-output	action sequence 输出	rule-result
chart/data/data-name	包含 pie slice 名称的数据列	REGION
chart/data/data-value	包含 pie slice 值的数据列	ACTUAL

chart/data/data-orientation	direction in which to process data to get the pie slices(rows/columns)	rows
-----------------------------	--	------

除了 widget XML 中定义的这些设置，调用者可指定这些设置：

设置	描述	值
drill-url	用于生成 drill 路径的一个 URL 模版。URL 可有包含在 '{' and '}' 中的可替换参数。	SampleDashboard.jsp?region={REGION}
Inner-param	在 URL 中包含将被替换的值的的数据列	REGION

chart 组件从这个文件中读取所有的设置，然后执行指定的 action sequence 为图表获取数据。sample dashboard 中的 action sequences 仅使用一个单一的组件生成数据，但是 BI 组件的任何序列可用来执行复杂的(例如 scripting 或 in-line ETL) 或远程的(例如 Web 服务) 任务。

4.2 Headcount Variance Bar Chart

这个 bar chart 的定义在：

pentaho-demo/pentaho-solutions/samples/dashboard/departments.widget.xml

这个 XML 文件定义了：

设置	描述	值
chart/type	图的类型(BarChart, LineChart, AreaChart)	BarChart
chart/title	图的标题	Headcount Variance
chart/title-position	图的标题的位置(上, 下, 左, 右)	top
chart/title-font/font-family	标题字体的名字	Ariel
chart/title-font/size	标题字体的大小	20
chart/title-font/is-bold	图的标题是粗体么?	false
chart/title-font/is-italic	图的标题是斜体么?	false
chart/is-3D	图是 3D 的么?	false
chart/border-visible	Does chart have a border?	false
chart/include-legend	Does chart have a legend	false
chart/chart-background	chart image 的背景。你可定义 gradients, textures, images 或 flat colors	type=color #FFFFFF
chart/plot-background	plot area 的背景	type=color #EEEEEE
chart/orientation	chart 的方向(垂直或水平)	horizontal
chart/is-stacked	Is chart stacked?	false
chart/color-palette/color	color to use for first series	#336699
chart/color-palette/color	color to use for second series	#99CCFF

chart/data/data-solution	solution for action sequence providing data for this chart	Samples
chart/data/data-path	action sequence 的路径	dashboard
chart/data/data-action	action sequence 的名字	regions_headcount_data.xaction
chart/data/data-output	action sequence 的输出	rule-result
chart/data/data-name	包含 pie slice names 的数据列	REGION
chart/data/data-value	包含 pie slice values 的数据列	ACTUAL
chart/data/data-orientation	direction in which to process data to get pie slices(rows/columns)	rows

除了在 widget XML 文件中定义的这些设置， caller 还指定了这些设置：

设置	描述	值
drill-url	一个 URL 模版用于生成 drill 路径。URL 可有包含在 '{' and '}' 中的可替换的参数	Dynamically set. e.g. SampleDashboard.jsp?region=Central &department={SERIES}
inner-param	data column containing values to be replaced in URL.	DEPARTMENT
REGION	在查询中用来选择数据的过滤器	Dynamically set. e.g. 'Eastern'
image-width	chart image 的宽度	450
image-height	chart image 的高度	300

REGION 参数被传送进 action sequence 用作一个查询查询过滤器。

4.3 动态Dial

这个 dial的定义在 pentaho-demo/pentaho-solutions/samples/dashboard/sampledial.widget.xml

这个 XML 文件定义了：

设置	描述	值
widget/dial/units	units of dial value	\$
widget/dial/plot-background/texture-image	dial area 的背景	/samples/portal/dial_03.gif
widget/dial/tick-interval	gap between ticks	5
widget/dial/tick-color	Ticks 的颜色	#808080
widget/dial/needle-color	Needle 的颜色	#808080
widget/dial/chart-background/texture-image	chart image 的背景	/samples/portal/dial_03.gif
widget/dial/interval/label	Interval name	'under'
widget/dial/interval/minimum	Interval lower limit	-15
widget/dial/interval/maximum	Interval upper limit	0
widget/dial/interval/color	Interval background color	#FFFFFF
widget/dial/interval/text-color	Interval text and tick color	#40BB40
widget/dial/interval/stroke-width	Interval line thickness	5

注意在文件中定义了两个 intervals: 一个覆盖了从 -15 到 0 的范围, 另一个是 0 到 +15。
除了在 widget XML 文件中定义的这些设置, caller 还指定了这些设置:

设置	描述	值
title	Dial 的标题	'My Dial'
value	Dial 上显示的值	Varies over time from -15 to +15
image-width	dial image 的宽度	105
image-height	dial image 的高度	105

In sample dashboard page creates a value for dial based on current time.

4.4 嵌入式报表

报表模板的定义在: pentaho-demo/pentaho-solutions/samples/dashboard/jsp/embedded_report.xml
报表模板被报表引擎(JFreeReport) 用于layout 嵌入式报表。你可使用 Pentaho Report Wizard 设计一个报表定义。

你可创建 HTML 的一个片段, 而不是一个完整的 HTML 页面, 你将需要编辑报表定义 XML 文件以在 report/configuration 的报表配置区域添加一个属性。

`<property name="org.jfree.report.modules .output.table.html .BodyFragment">true</property>`
为将报表内容中的一个 drill link 添加进一个 URL, 你需要编辑报表定义, 并添加一个 URL 表达式, 例如 sample dashboard 中的这个。

```
<expression name="URLCreateExpression"
class="org.jfree.report. function.TextFormatExpression">
<properties>
<property name="pattern"
>SampleDrill.jsp?region={ 0} &position={ 1} &department={ 2} </property>
<property name="field[ 0] ">REGION</property>
<property name="field[ 1] ">POSITIONTITLE</property> <property name="field[ 2] ">DEPARTMENT</property>
</properties>
</expression>
```

5 Action Sequences

5.1 Headcount Spending By Region Action Sequence

这个 action sequence 的定义在:

pentaho-demo/pentaho-solutions/samples/dashboard/regions_headcount_data.xaction

你可以在 Eclipse IDE 中使用 Pentaho Design Studio 以编辑这个文件。

这个 action sequence 包含一个 SQL Query Rule, 其从 Pentaho demo 服务器提供的 sample 数据库获取一些数据。action sequence 不带任何参数。SQL Query Rule 执行这个查询:

```
select REGION, sum(ACTUAL) as ACTUAL
from quadrant _actuals
group by REGION order by ACTUAL
```

并在一个名为 'rule-result' 的输出中返回数据。这个数据用于生成 'Headcount Spending By Region' 饼图。

5.2 Headcount Variance Action Sequence

这个 action sequence 的定义在:

pentaho-demo/pentaho-solutions/samples/dashboard/department_variance_data.xaction

你可以在 Eclipse IDE 中使用 Pentaho Design Studio 以编辑这个文件。这个 action sequence 包含一个 SQL Query Rule, 其从 Pentaho demo 服务器提供的 sample 数据库获取一些数据。

action sequence 带有一个参数 REGION, 其在查询中用作过滤器, SQL Query Rule 执行这个查询:

```
select department, sum(variance) from quadrant _actuals
where region='{REGION}'
group by department
```

在名为 'rule-result' 的输出中返回数据。这个数据用于生成 'Headcount Variance By Region' 饼图。

5.3 Embedded Report Action Sequence

这个报表的定义在:

pentaho-demo/pentaho-solutions/samples/dashboard/jsp/embedded_report.xaction

你可以在 Eclipse IDE 中使用 Pentaho Design Studio 以编辑这个文件。这个 action sequence 包含一个 JFree Report 组件, 其从 sample 数据库获取数据, 使用一个报表模版生成一个 HTML 片段。

action sequence 带有参数 'region' 和 'department', 其用作查询中的过滤器。报表组件执行这个查询:

```
select QUADRANT_ACTUALS.REGION, QUADRANT_ACTUALS.DEPARTMENT, QUADRANT
_ACTUALS.POSITIONTITLE,
QUADRANT_ACTUALS.ACTUAL, QUADRANT_ACTUALS.BUDGET, QUADRANT_ACTUALS.VARIANCE
from QUADRANT_ACTUALS
where QUADRANT_ACTUALS.REGION = '{region}'
and QUADRANT_ACTUALS.DEPARTMENT = '{department}'
order by QUADRANT_ACTUALS.REGION, QUADRANT_ACTUALS.DEPARTMENT
and uses a report template
```

pentaho-demo/pentaho-solutions/samples/dashboard/jsp/embedded_report.xml (参考上面的嵌入式报表)

报表组件在一个名为 'report' 的输出中返回 HTML, action sequence 将这个输出转向到 response 的 output stream。

6 附录1: Dashboard JSP

这是 JSP Sample Dashboard 的源代码:

```
<%Q page language="java"
import="java . util .ArrayList,
java .util .Date,
java.io . ByteArrayOutputStream,
org.pentaho . core . ui . SimpleUrlFactory,
org.pentaho .messages .Messages,
org.pentaho . core . system . PentahoSystem,
org.pentaho . ui .component . DashboardWidgetComponent,
org.pentaho .core . solution . HttpRequestParameterProvider, org.pentaho .core . solution . HttpSessionParameterProvider,
org.pentaho.core . session . IPentahoSession,
org.pentaho .core . util .UIUtil,
org.pentaho . util . VersionHelper,
org.pentaho .messages . util . LocaleHelper,
org.pentaho .core . solution . ActionResource,
org.pentaho .core . solution . IActionResource,
```

```
org.pentaho.core.solution.SimpleParameterProvider, org.pentaho.ui.ChartHelper,  
java.io.*"  
%><%  
/*
```

Copyright 2006 Pentaho Corporation. All rights reserved.

This software was developed by Pentaho Corporation and is provided under the terms

```
<tr> of the Mozilla Public License, Version 1.1, or any later version. You may not use  
this file except in compliance with the license. If you need a copy of the license,  
please go to http://www.mozilla.org/MPL/MPL-1.1.txt. The Original Code is the Pentaho  
BI Platform. The Initial Developer is Pentaho Corporation.  
*
```

```
<%  
Software distributed under the Mozilla Public License is distributed on an "AS IS"  
basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. Please refer to  
the license for the specific language governing your rights and limitations.  
*
```

```
Created Feb 16, 2006
```

```
Qauthor James Dixon
```

```
*/
```

```
/*
```

```
This JSP is an example of how to use Pentaho components to build a dashboard.  
The script in this file controls the layout and content generation of the dashboard.  
See the document 'Dashboard Builder Guide' for more details
```

```
*/
```

```
// set the character encoding e.g. UTF-8
```

```
response.setCharacterEncoding(LocaleHelper.getSystemEncoding());
```

```
// create a new Pentaho session
```

```
IPentahoSession userSession = UIUtil.getPentahoSession(request);
```

```
%>
```

```
<html>
```

```
<head>
```

```
<title>Pentaho Sample Dashboard - JSP</title>
```

```
</head> <body>
```

```
<%
```

```
// See if we have a 'department' parameter
```

```
String department = request.getParameter("department"); // See if we have a 'region' parameter
```

```
String region = request.getParameter("region");
```

```
// Create the title for the top of the page
```

```
String title = "Select a region";
```

```
if( department != null ) {
```

```
title = "This is headcount spending for " + region + ", " + department;
```

```
}
```

```
else if ( region != null ) {
```

```
title = "This is headcount spending for " + region;
```

```
} %>
```

```
<h1 style='font-family:Arial'><%= title %></h1>
```

```
<table>
```

```
<td valign="top">
```

```
<span style="font-family:Arial;font-weight:bold"> Select a Region By Clicking on the Pie Chart </span>
```

```
// Make a pie chart showing the regions
```

```
// create the parameters for the pie chart
```

```
SimpleParameterProvider parameters = new SimpleParameterProvider(); // define the click url template
```

```
parameters.setParameter("drill-url", "SampleDashboard.jsp?region={ REGION} "); // define the slices of the pie chart
```

```
parameters.setParameter("inner-param", "REGION");
```

```
// set the width and the height
```

```
parameters.setParameter("image-width", "450");
```

```

parameters. setParameter( "image-height", "300");
StringBuffer content = new StringBuffer();
ArrayList messages = new ArrayList();
// call the chart helper to generate the pie chart image and get the HTML content // use the chart definition in
'samples/dashboard/regions.widget.xml' ChartHelper.doPieChart ( "samples", "dashboard", "regions .widget.xml",
parameters, content, userSession, messages, null );
%>
<%= content.toString() %>
</td>
<td valign="top"><span style="font-family:Arial;font-weight:bold">
<%
if( region != null ) {
// if the user has clicked on a slice of the pie chart
// we should have a region to work with
%>
Select a Department By Clicking on the Bar Chart
<%
// Make a bar chart showing the department
// create the parameres for the bar chart
parameters = new SimpleParameterProvider();
// define the click url template
parameters. setParameter ( "drill-url",
"SampleDashboard.jsp? region="+region+"&department={ SERIES } " ); parameters.setParameter( "REGION", region );
parameters.setParameter( "outer-params", "REGION" );
// define the category axis of the bar chart
parameters.setParameter( "inner-param", "DEPARTMENT");
// set the width and the height
parameters. setParameter( "image-width", "450");
parameters. setParameter( "image-height", "300");
content = new StringBuffer();
messages = new ArrayList();
// call ChartHelper to create the pie chart image and get the HTML content
// use the chart definition in 'samples/dashboard/regions.widget.xml' ChartHelper.doChart ( "samples", "dashboard",
"departments .widget.xml", parameters, content, userSession, messages, null );
%>
</span>
<br/>
<%= content.toString() %>
<%
<td colspan="2" valign="top" style="font-family:Arial;font-weight:bold"> <hr size="1"/>
</td>
<td valign="top"><span style="font-family:Arial;font-weight:bold">
if( department != null ) {
// if the user has clicked on a bar of the bar chart // we should have a region and department to work with
// create a dial and supply a value we create from the current time
Date now = new Date();
int seconds = now.getSeconds(); // create a value from -15 to +15 int dialValue = -15+seconds/2;
// create the parameres for the bar chart
parameters = new SimpleParameterProvider();
// set the value displayed on the dial
parameters.setParameter( "value", ""+dialValue );
// set the title for the dial parameters.setParameter( "title", "My Dial" );
// set the width and the height
parameters. setParameter( "image-width", "105");
parameters. setParameter( "image-height", "105");

```

```

content = new StringBuffer(); messages = new ArrayList();
// call ChartHelper to create the pie chart image and to the HTML content
// use the chart definition in 'samples/dashboard/regions.widget.xml' ChartHelper.doDial ( "samples", "dashboard",
"samplodial .widget.xml",
parameters, content, userSession, messages, null );
</span>
<br/>
<%= content.toString() %>
}
</td>
<td valign="top" style="font-family:Arial;font-weight:bold">
if( department != null ) {
// if the user has clicked on a bar of the bar chart // we should have a region and department to work with
// run a report and embed the content into this page
// create the parameres for the report
parameters = new SimpleParameterProvider();
// pass the region and department to the report
parameters.setParameter( "region", region );
parameters.setParameter( "department", department );
// create an output stream for the report content
ByteArrayOutputStream outputStream = new ByteArrayOutputStream(); messages = new ArrayList();
// run the action sequence 'samples/dashboard/jsp/report.xaction' ChartHelper.doAction( "samples", "dashboard/jsp",
"embedded_report .xaction", "SampleDashboard. jsp", parameters, outputStream , userSession, messages, null ); // write the report
content into this page
out.write( outputStream.toString() );
}
</td>
</ tr> </table>
</body> </html>

```

7 附录2: Pie Chart Definition

```

<chart>
<!-- This file defines the pie chart that shows
the actual values for each region -->
<!-- Specify the title of the chart --> <title>Headcount Spending by Region</title>
<!-- Specify the location of the title --> <title-position>none</title-position>
<!-- Specify the font of the title --> <title-font>
<font-family>Ariel</font-family> <size>20</size>
<is-bold>>false</is-bold>
<is-italic>>false</is-italic> </title-font>
<width> 450</width> <height>300</height>
<!-- Specify the 3D-ness of the bars --> <is-3D> false </ is-3D>
<!-- Specify if the chart has a border and the border color --> <border-visible> false </border-visible>
<border-paint> #bbbbff </border-paint>
<!-- Specify is the chart legend should be shown --> <include-legend>false</include-legend>
<!-- Specify where the data for the chart comes from -->
<data>
<!-- Specify the path to the action sequence that provides the data --> <data-solution>samples</data-solution>
<data-path>dashboard</data-path>
<data-action>regions_headcount_data .xaction</data-action>
<!-- Specify the output of the action sequence that contains the data --> <data-output>rule-result</data-output>
<data-name>POSITIONTITLE</data-name>
<data-value>ACTUAL</data-value>

```

```

<!-- Specify whether to get the pie series from the rows or columns --> <data-orientation>columns< /data-orientation>
</data>
</chart>

```

8 附录3: Region Data

```

<action-sequence>
<version> 1</version>
<title>Regions and departments</title>
<logging-level>debug< /logging-level>
<documentation>
<author>James Dixon< /author>
<description>
Return the actual headcount costs total for each region
</description> <help>< /help> </documentation>
<inputs/>
<!-- Define an output called 'rule-result' --> <outputs>
<rule-result>
<type>list</type>
</rule-result>
</outputs>
<!-- This action sequence does not require any external resources --> <resources/>
<actions>
<action-definition>
<action-inputs/>
<!-- Define a local output called 'rule-result' --> <action-outputs>
<rule-result type="list"/>
</action-outputs>
<!-- Specify the component to execute --> <component-name> SQLLookupRule< /component-name> <action-type>rule<
/ action-type>
<!-- Define the settings for the component --> <component-definition>
<!-- Define the datasource for the query -->
<j ndi> SampleData< /j ndi>
<!-- Define the query to execute.
Note the parameter {REGION} in the query --> <query><![CDATA[
select REGION, sum(ACTUAL)
from quadrant _actuals
group by REGION
order by ACTUAL]]>
</query>
</component-definition> </action-definition>
</actions>
</action-sequence>

```

9 附录4: Department Bar Chart Definition

```

<chart>
<!-- This file defines the bar chart that shows the
actual-to-budget variance for each department -->
<!-- Define the chart type --> <chart-type>BarChart< /chart-type>
<!-- Specify the title of the chart --> <title>Headcount Variance</title>
<!-- Specify the location of the title --> <title-position>TOP</title-position>
<!-- Specify the font of the title --> <title-font>
<font-family>Ariel</font-family> <size>20</size>

```

```

</is-bold>>false</is-bold>
</is-italic>>false</is-italic> </title-font>
<chart-background-color> #FF80FF< /chart-background-color> <plot-background-color> #FFFF00< /plot-background-color>
<!-- Specify the orientation of the bars --> <orientation>Horizontal< /orientation>
<!-- Specify the 3D-ness of the bars --> <is-3D> false< /is-3D>
<!-- Specify if the bars are stacked --> <is-stacked>>false</is-stacked>
<!-- Specify if the chart has a border and the border color --> <border-visible>>true< /border-visible>
<border-paint> #000000</border-paint>
<!-- Specify if the chart legend should be shown --> <include-legend>>true< /include-legend>
<!-- Specify the color palette for the chart -->
<color-palette>
<color>#336699</color> <color>#99CCFF</color> <color>#999933</color> <color>#666699</color>
<color>#CC9933</color> <color>#00 6666</color> <color>#3399FF</color> <color>#993300</color>
<color>#CCCC99</color> <color># 666666</color> <color>#FFCC66</color> <color>#6699CC</color>
<color>#663366</color>
</color-palette>
<!-- Specify where the data for the chart comes from -->
<data>
<!-- Specify the path to the action sequence that provides the data --> <data-solution>samples</data-solution>
<data-path>dashboard< /data-path>
<data-action>department_variance_data .xaction</data-action>
<!-- Specify the output of the action sequence that contains the data --> <data-output>rule-result< /data-output>
<!-- Specify whether to get the chart series from the rows or columns --> <data-orientation>rows< /data-orientation>
</data>
</chart>

```

10附录5: Department Variance Data

这个 action sequence 执行一个 SQL 查询，获得一个指定区域内每个部门的 headcount actual 和 budget 的不同。这个 action sequence 的定义在 ~/pentaho-demo/pentaho-solutions/samples/dashboard/department_variance_dial.xaction.

Use the Pentaho BI Studio

```

<?xml version="1.0" encoding="UTF-8"?> <action-sequence>
<version> 1</version>
<title>Regions and departments</title> <logging-level>debug< /logging-level> <documentation>
<author>James Dixon< /author>
<description>Return the variance between headcount actual and budget
for every department in the specified region
</description>
<help>just testing. . .</help> </documentation>
<!-- Define an input called 'REGION'.
This will be passed in when the user clicks on a slice of the pie chart
-->
<inputs>
<REGION type="string">
<sources>
<request>REGION< /request>
</sources>
</REGION>
</inputs>
<!-- Define an output called 'rule-result' --> <outputs>
<rule-result>
<type>list</type>

```

```

</rule-result>
</outputs>
<!-- This action sequence does not require any external resources --> <resources/>
<actions>
<action-definition>
<!-- Define a local input called 'REGION' -->
<action-inputs>
<REGION type="string"/> </action-inputs>
<!-- Define a local output called 'rule-result' --> <action-outputs>
<rule-result type="list"/>
</action-outputs>
<!-- Specify the component to execute --> <component-name> SQLLookupRule< /component-name> <action-type>rule<
/action-type>
<!-- Define the settings for the component --> <component-definition>
<!-- Define the datasource for the query --> <j ndi> SampleData< /j ndi>
<!-- Define the query to execute.
Note the parameter {REGION} in the query
-->
<query><![CDATA[
select department, sum(variance)
from quadrant _actuals
where region='{REGION}'
group by department
]]>
</query>
</component-definition> </action-definition>
</actions>
</action-sequence>

```

11 附录6: Dial Definition

```

<widget>
<dial>
<name>Gauge 1</name> <units>$/units>
<!-- this is the background for the whole image --> <!-- background-color>#ffffff</background-color -->
<!-- this ia the background for the dial -->
<!-- plot-background-color>#777700</plot-background-color -->
<plot-background type="texture">
<texture-image>/samples/portal/dial_03 .gif</texture-image> </plot-background>
<tick-interval>5</tick-interval>
<value-color>#9999bb</value-color> <tick-color>#808080</tick-color>
<!-- this is the color of the needle --> <needle-color>#808080</needle-color>
<chart-background type=" texture">
<texture-image>/samples/portal/dial_03 .gif</texture-image> </chart-background>
<!-- intervals define ranges on the dial that are
colored differently from the dial background -->
<interval>
<label>under</label>
<!-- this is the value that the range starts at --> <minimum>-15</minimum>
<!-- this is the value that the range stops at -->
<maximum> 0</maximum>
<!-- this is the color of the range -->
<color> # ffffff< / color>
<!-- this is the color of the text for the

```

```

range value and tick marks --> <text-color>#40bb40</text-color>
<stroke-width> 5</stroke-width>
</interval>
<interval>
<label>over</label>
<minimum> 0</minimum>
<maximum> 15</maximum>
<color> # ffffff< / color>
<text-color>#bb4040</text-color> <stroke-width> 5</stroke-width> </interval>
</dial> </widget>

```

12附录7: Embedded Report Template

```

<?xml version="1.0" encoding="UTF-8"?> <action-sequence>
<version> 1</version>
<title>JFreeReport HTML Example</title> <logging-level>debug< /logging-level> <documentation>
<author>James Dixon< /author> <description><![CDATA[
This is an example of an HTML report produced by JFreeReport.
<p>It shows the actual headcount cost, budgeted headcount
cost, and variance for every position in the specified
department and region ] ] ></description>
<icon>/style/icons/j free1 .png</icon>
<help>< /help>
</documentation>
<!-- Define an input called 'region' and an input called 'department'.
These will be passed in when the user clicks on a bar on the bar chart --> <inputs>
<region type="string"> <sources>
<request>region< /request>
</sources> </region>
<department type="string">
<sources>
<request>department< /request>
</sources>
</department>
</inputs>
<!-- Define an output called 'report' -->
<outputs>
<report type="content">
<destinations>
<response>content< /response> </destinations>
</report>
</outputs>
<!-- This action sequence uses a report definition file embedded_report.xml -->
<resources>
<report-definition>
<solution-file>
<location>embedded_report.xml</location> <mime-type> text/xml< /mime-type>
</solution-file>
</report-definition>
</resources>
<actions>
<action-definition>
<!-- Define a local inputs called 'region' and 'department' --> <action-inputs>
<region type="string"/>

```

```

<department type="string"/>
</action-inputs>
<!-- Define a local output called 'report' --> <action-outputs>
<report type="content"/>
</action-outputs>
<!-- Specify the component to execute --> <component-name>JFreeReportComponent</component-name> <action-type>report<
/action-type>
<!-- Define the settings for the component --> <component-definition>
<live>>false</live>
<!-- Define the datasource for the query --> <j ndi> SampleData</j ndi>
<source> sql</source>
<!-- Define the query to execute. Note the
parameter {region} and {department} in the query --> <query><![CDATA[
select QUADRANT_ACTUALS . REGION, QUADRANT_ACTUALS . DEPARTMENT, QUADRANT_ACTUALS .
POSITIONTITLE,
QUADRANT_ACTUALS . ACTUAL, QUADRANT_ACTUALS . BUDGET, QUADRANT_ACTUALS . VARIANCE
from QUADRANT_ACTUALS
where QUADRANT_ACTUALS.REGION = '{region}'
and QUADRANT_ACTUALS.DEPARTMENT = '{department}' order by QUADRANT_ACTUALS.REGION,
QUADRANT_ACTUALS.DEPARTMENT]]>
</query>
<!-- Define the content type for the report --> <output-type>html</output-type>
</component-definition>
</action-definition>
</actions>
</action-sequence>

```

13附录8: Drill JSP

```

<%Q page language="java"
import="java.util.ArrayList,
java.util.Date,
java.io . ByteArrayOutputStream,
org.pentaho . core . ui . SimpleUrlFactory,
org.pentaho .messages .Messages,
org.pentaho . core . system . PentahoSystem,
org.pentaho . ui .component . DashboardWidgetComponent,
org.pentaho .core . solution . HttpRequestParameterProvider, org.pentaho .core . solution . HttpSessionParameterProvider,
org.pentaho .core . session . IPentahoSession,
org.pentaho .core . util . UIUtil,
org.pentaho . util . VersionHelper,
org.pentaho .messages . util . LocaleHelper,
org.pentaho .core . solution . ActionResource,
org.pentaho .core . solution . IActionResource,
org.pentaho .core . solution . SimpleParameterProvider, org.pentaho . ui . ChartHelper,
java.io .*"
%><%
/*

```

Copyright 2006 Pentaho Corporation. All rights reserved.

This software was developed by Pentaho Corporation and is provided under the terms of the Mozilla Public License, Version 1.1, or any later version. You may not use this file except in compliance with the license. If you need a copy of the license, please go to <http://www.mozilla.org/MPL/MPL-1.1.txt>. The Original Code is the Pentaho BI Platform. The Initial Developer is Pentaho Corporation.

*

Software distributed under the Mozilla Public License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. Please refer to the license for the specific language governing your rights and limitations.

*

Created Feb 16, 2006

Qauthor James Dixon

*/

/*

This JSP page is part of the Pentaho samples that show how JSP can be used to present and control content that is generated by the Pentaho BI Platform.

This JSP page is the page that is used after a user clicks on a link in the embedded report that is displayed in SampleDashboard.jsp. The region, department, and job position title are passed in as parameters on the URL

The url is formatted in the report definition file:

~/pentaho-demo/pentaho-solutions/samples/dashboard/j sp/embedded_report.xml

The url is set in the last function that is defined in the file. The important line is the "pattern"

```
<expression name="URLCreateExpression"
```

```
class="org.jfree.report.function.TextFormatExpression">
```

```
<properties>
```

```
<property
```

```
name="pattern">SampleDrill.jsp?region={ 0 } &amp;position={ 1 } &amp;department={ 2 } </property> <property name="field[ 0]">REGION</property>
```

```
<property name="field[ 1]">POSITIONTITLE</property>
```

```
<property name="field[ 2]">DEPARTMENT</property>
```

```
</properties> </expression>
```

You can change the url template to point to wherever you need it to

```
String region = request.getParameter("region");
```

```
String department = request.getParameter("department"); String position = request.getParameter("position");
```

```
%>
```

```
<html>
```

```
<head>
```

```
<title>Pentaho Regional Report - JSP Sample</title> </head>
```

```
<body>
```

```
<h1 style="font-family:Arial">Drill Destination</h1>
```

```
<span style="font-family:Arial">
```

```
The selected region is '<%= region %>'
```

```
<p/>
```

```
The selected department is '<%= department %>'
```

```
<p/>
```

```
The selected position is '<%= position %>'
```

```
</span>
```

```
</body> </html>
```