JBoss BIRT Integration User Guide

Version: 1.1.1.GA

1. Introduction 1
1.1. What is BIRT? 1
1.2. JBoss BIRT Integration Functionality Overview 1
2. Tasks 3
2.1. Adding BIRT Functionality to Standard Seam Web Project 3
2.1.1. Creating Seam Web Project with Birt Facet 3
2.1.2. Integration with Seam 8
2.2. Using Hibernate ODA Data Source 11
2.3. Creating a Dynamic BIRT Report 18
3. JBoss BIRT Integraion Framework API Reference 25
3.1. <b:birt> Component 25
3.2. <b:param> Component 27
4. Other Relevant Resources on the Topic 29

Introduction

1.1. What is BIRT?

BIRT plugin is an Eclipse-based open source reporting system for web applications based on Java and J2EE. BIRT consists of two main components: a report designer based on Eclipse, and a runtime component that can be added to your app server. BIRT also offers a charting engine that lets you add charts to your own Web application.

With the help of BIRT you can generate a great number of reports to be used in your application:

- Lists
- Charts
- Crosstabs, or cross-tabulation, or matrix
- Letters and Documents
- Compound Reports

You can find more detailed information on the BIRT plugin, its report types and anatomy on the *BIRT Homepage* [http://www.eclipse.org/birt/phoenix/intro/].

To understand the basic BIRT concepts and to know how to create a basic BIRT report, refer to the *Eclipse BIRT Tutorials* [http://www.eclipse.org/birt/phoenix/tutorial/]. What extensions JBoss Tools provides for Eclipse BIRT you'll find out in the next sections.

1.2. JBoss BIRT Integration Functionality Overview

The key feature of JBoss BIRT Integration is the JBoss BIRT Integration Framework, which allows to integrate a BIRT report into Seam/JSF container. The framework API reference is in the *JBoss BIRT Integraion Framework API Reference* chapter of the guide.

This guide also covers functionality of JBoss Tools module which assists in integration with BIRT. The integration plug-in allows you to visually configure Hibernate Data Source (specify a Hibernate configuration or JNDI URL), compose HQL queries with syntax-highlighting, content-assist, formatting as well as other functionalities available in the HQL editor.

To enable JBoss Tools integration with BIRT you are intended to have the next:

- Eclipse with JBoss Tools installed (how to install JBoss Tools on Eclipse, what dependences and versions requirements are needed reed in the JBoss Tools Installation section)
- BIRT Report Designer (BIRT Report Designer 2.3.2 you can download from *Eclipse downloads site* [http://www.eclipse.org/downloads/download.php?file=/birt/downloads/drops/R-R1-2_3_2-200902181355/birt-report-framework-2_3_2.zip])

- BIRT Web Tools Integration (BIRT WTP Integration 2.3.2 you can download from *Eclipse downloads site* [http://www.eclipse.org/downloads/download.php?file=/birt/downloads/drops/R-R1-2_3_2-200902181355/birt-wtp-integration-sdk-2_3_2.zip])

Note:

Versions of BIRT framework and BIRT WTP integration should be no less than RC4 in order to the BIRT facet works correctly.

Tasks

In this chapter of the guide you will find information on the tasks that you can perform integrating BIRT. The required version of BIRT is 2.3.2 or greater.

2.1. Adding BIRT Functionality to Standard Seam Web Project

In this section you'll know how to integrate BIRT into a Seam web project.

You are supposed to have Seam runtime and JBoss Application Server downloaded and extracted on your hard drive. You can download Seam from the *Seam Framework* [http://seamframework.org/Download] web page and JBoss Application Server from *JBoss Application Server* [http://jboss.org/jbossas/downloads/] official site.

Tip:

We used *JBoss Seam 2.0.1 GA* [http://sourceforge.net/project/showfiles.php?group_id=22866&package_id=163777] and *JBoss Application Server 4.2.2 GA* [http://sourceforge.net/project/showfiles.php?group_id=22866&package_id=163777] in examples of this guide.

2.1.1. Creating Seam Web Project with Birt Facet

We recommend to open Seam perspective by going to *Window* > *Open Perspective* > *Other* > *Seam* , this way you will have all the tools to work with Seam at hand. To create a new Seam Web project navigate to *File* > *New* > *Seam Web Project* . Otherwise you should go to *File* > *New* > *Other* > *Seam* > *Seam Web Project* if Seam perspective is not active.

1. On the first wizard page enter the project name, then specify the target runtime and target server. We recommend to use JBoss AS server and runtime environment to ensure best performance.

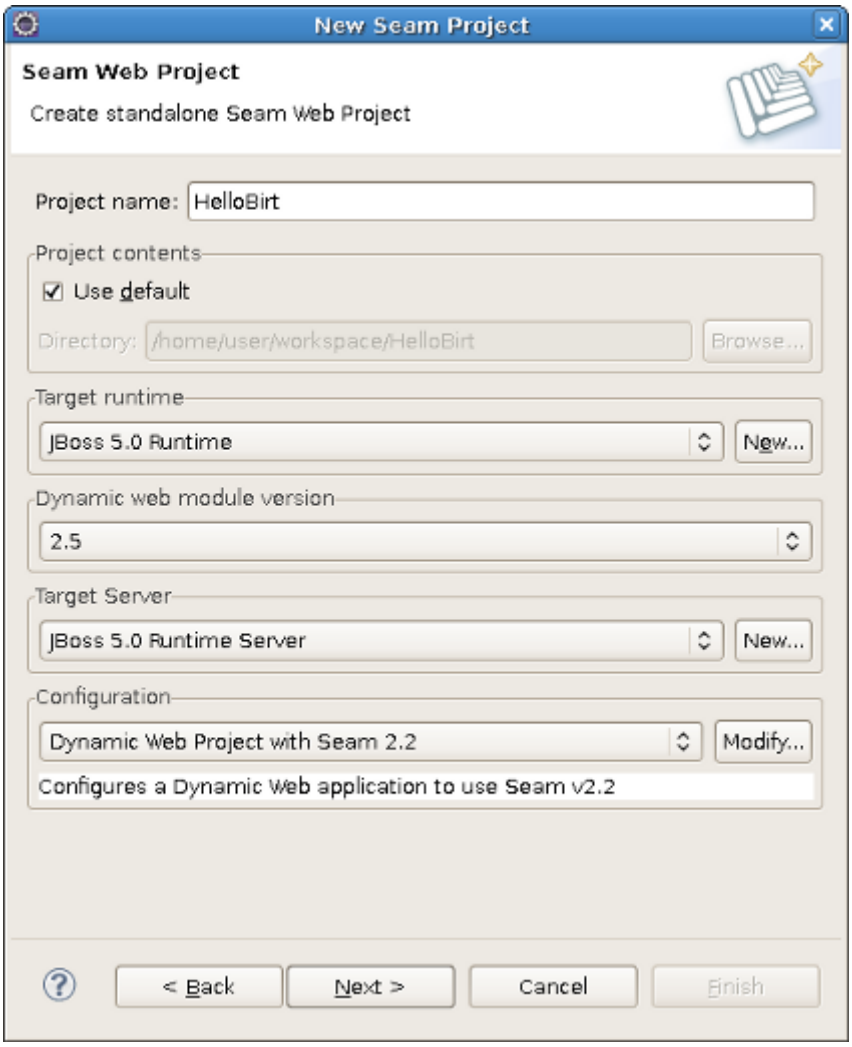

Figure 2.1. Creating Seam Web Project

2. In the *Configuration* group choose the version of *Seam* framework you are planning to use in your application. In this guide we used Seam 2.2.

3. Click the *Modify* button and enable the *Birt Reporting Runtime Component* facet by checking the appropriate option.

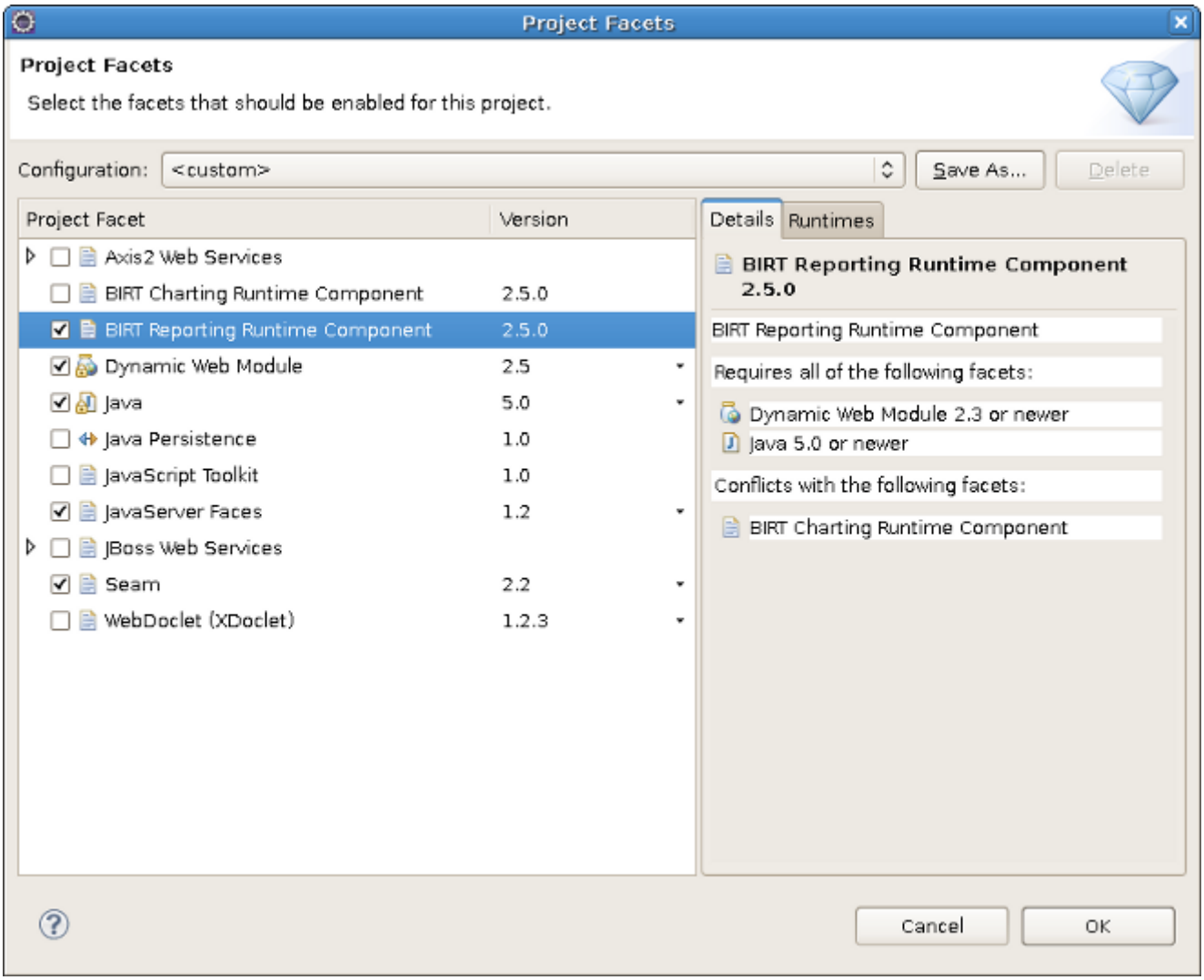

Figure 2.2. Adding the Birt Reporting Runtime Component Facet

Alternatively you can just choose the *JBoss BIRT Integration Web Project* configuration from the drop-down list in the *Configuration* group

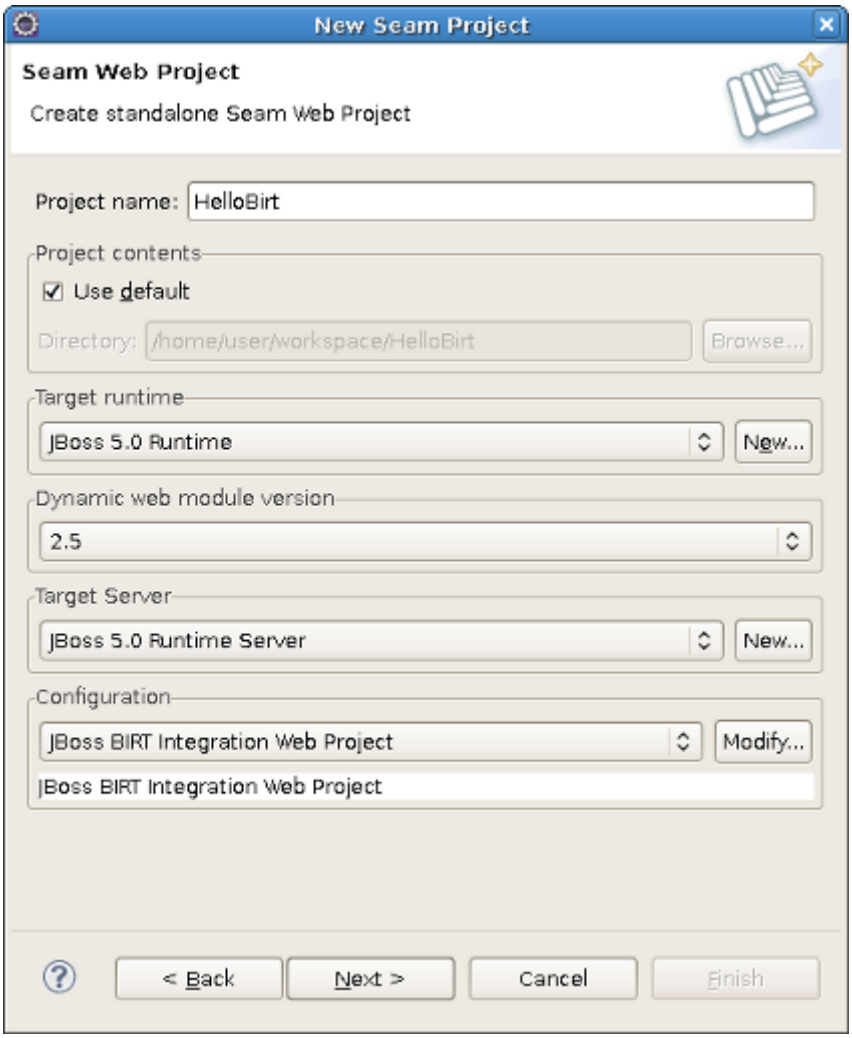

Figure 2.3. Choosing the JBoss BIRT Integration Web Project Configuration

4. You may leave the next two pages with default values, just press *Next* to proceed.

5. On the *Birt Configuration* page you can modify the BIRT deployment settings, which you can edit afterwards in the web.xml of the generated project. Let's keep the default values for now.

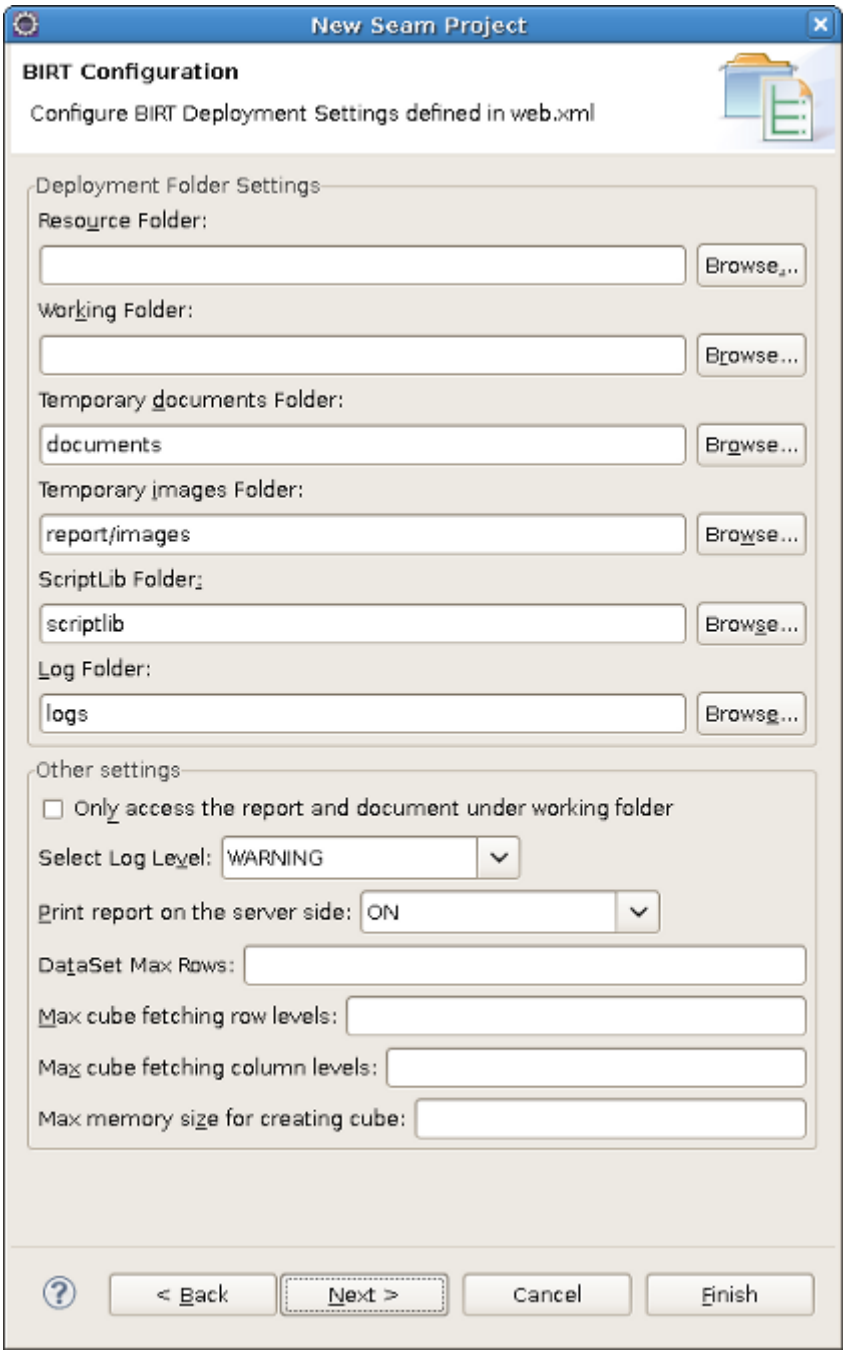

Figure 2.4. Configuring BIRT Deployment Settings

6. You can also leave the *JSF Capabilities* page with default values.

7. On the *Seam Facet* page you should specify the Seam runtime and Connection profile. Please note, that the Seam runtime must be of the version you initially specified in the project settings (See the *Creating Seam Web Project* figure).

   When creating a Seam project with BIRT capabilities you can use the BIRT Classic Models Sample Database connection profile to work with the BIRT sample database.

For more details on how to configure database connection for a Seam project please read the *Configure Seam Facet Settings* [http://docs.jboss.org/tools/3.0.0.GA/en/seam/html_single/index.html#seamFacet] chapter of Seam Dev Tools Reference Guide.

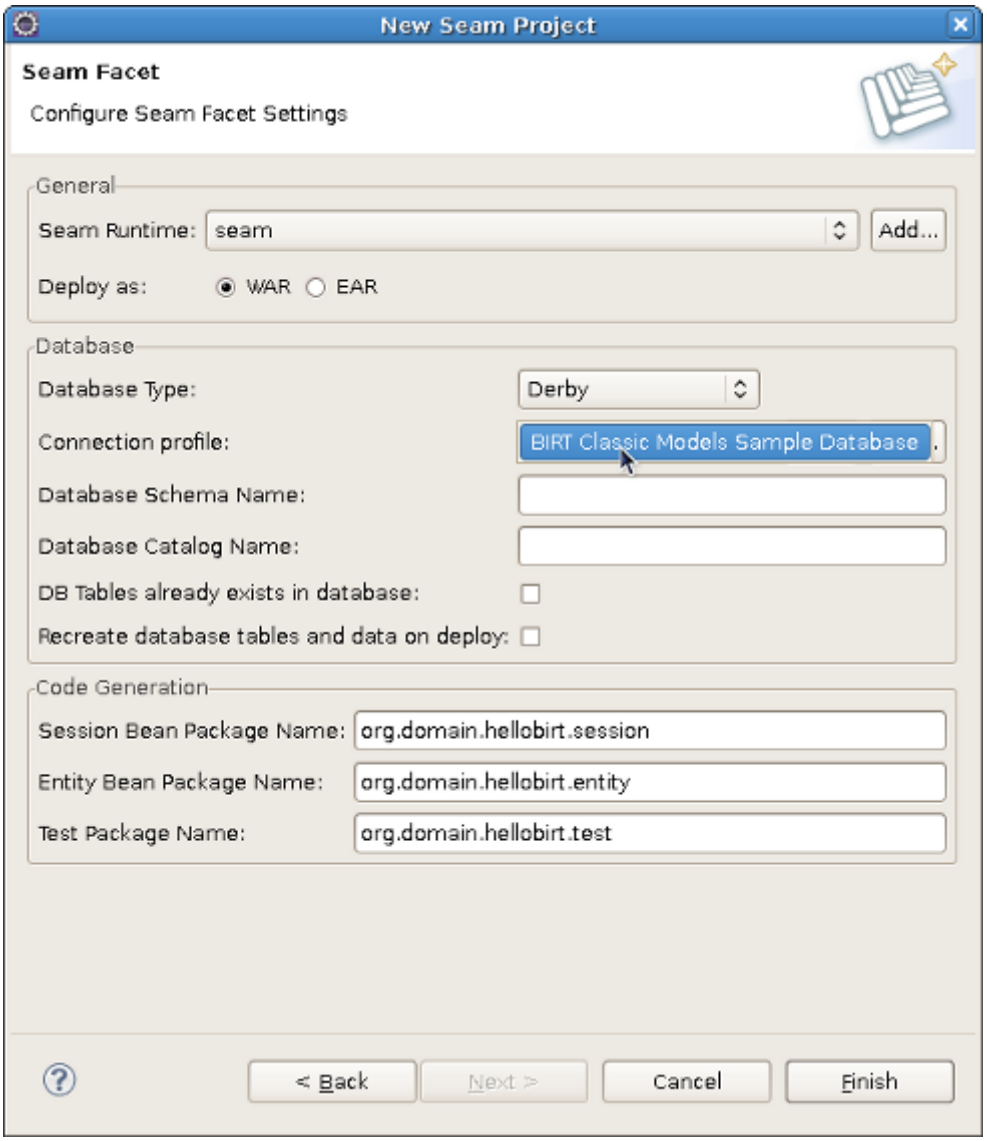

Figure 2.5. Configuring the Seam Facet settings

8. Hit *Finish* to create the project with BIRT functionality enabled.

2.1.2. Integration with Seam

In the previous section you have created a Seam project with BIRT capabilities. Now you can create a simple kick start project to see that everything is configured correctly.

1. Now create a BIRT report file and insert test data into the file. Name the report file helloBirt.rptdesign, the report should print the data from the

`CLASSICMODELS.CUSTOMERS` table of the BIRT Classic Models Sample Database, namely: customer number (`CLASSICMODELS.CUSTOMERS.CUSTOMERNAME`), contact person first name (`CLASSICMODELS.CUSTOMERS.CONTACTFIRSTNAME`) , contact person last name (`CLASSICMODELS.CUSTOMERS.CONTACTLASTNAME`) and contact person phone number(`CLASSICMODELS.CUSTOMERS.PHONE`).

The title of the report should be set via `reportTitle` parameter

As this guide is primarily focused on the BIRT integration and not the BIRT technology itself we will not show the steps required to make the report. If you do not feel strong about creating a BIRT report file please read *BIRT documentation* [http://eclipse.org/birt/phoenix/tutorial/].

You can download the file helloBirt.rptdesign *here* [files/helloBirt.rptdesign] and copy it to the WebContent folder of your Seam project.

2. When you are done with the helloBirt.rptdesign file, you should create a .xhtml file that will contain the BIRT report you have just created.

   The JBoss BIRT Integration framework provides 2 components represented as **<b:birt>** and **<b:param>** tags. The jboss-seam-birt.jar library implements the functionality of the components. To find more information about the framework pleas read the *JBoss BIRT Integraion Framework API Reference* chapter. To use that tags on the page you need to declare the tag library and define the name space like this:

```
xmlns:b="http://jboss.com/products/seam/birt"
```

   The **<b:birt>** is a container for a BIRT report, that helps you integrate the report into Seam environment. You can manage the properties of the report using the attributes of the **<b:birt>** tag.

   The **<b:param>** tag describes report parameters. To set a parameter you need to specify it's name the value you want to pass. You can use EL expressions to bind the representation layer with back-end logic.

3. Create the helloBirt.xhtml file in the WebContent with the following content:

```
...
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
 xmlns:s="http://jboss.com/products/seam/taglib"
 xmlns:ui="http://java.sun.com/jsf/facelets"
 xmlns:rich="http://richfaces.org/rich"
 xmlns:b="http://jboss.com/products/seam/birt"
```

```
 template="layout/template.xhtml">
 <ui:define name="body">
  <rich:panel>
   <b:birt designType="embed" designName="helloBirt.rptdesign"
    embeddable="true">
    <b:param name="reportTitle" value="Customers Contacts" />
   </b:birt>
  </rich:panel>
 </ui:define>
</ui:composition>
...
```

From this listing above you see that the title of the report is set via **<b:param>** by setting the parameter name and defining the *"value"* attribute with the "Customers Contacts" value.

4. We have created a Seam project and inserted the helloBirt report into the helloBirt.xhtml view file.

   To see that the application works correctly and as you expect, you need to launch it on the server.

5. In the Servers view (If it is not open navigate to Windows > Show View > Other > Server > Servers), select the server the application is deployed to and hit the Start button.

6. When the server is started, open your favourite browser and point it to `http://localhost:8080/HelloBirt/helloBirt.seam`.

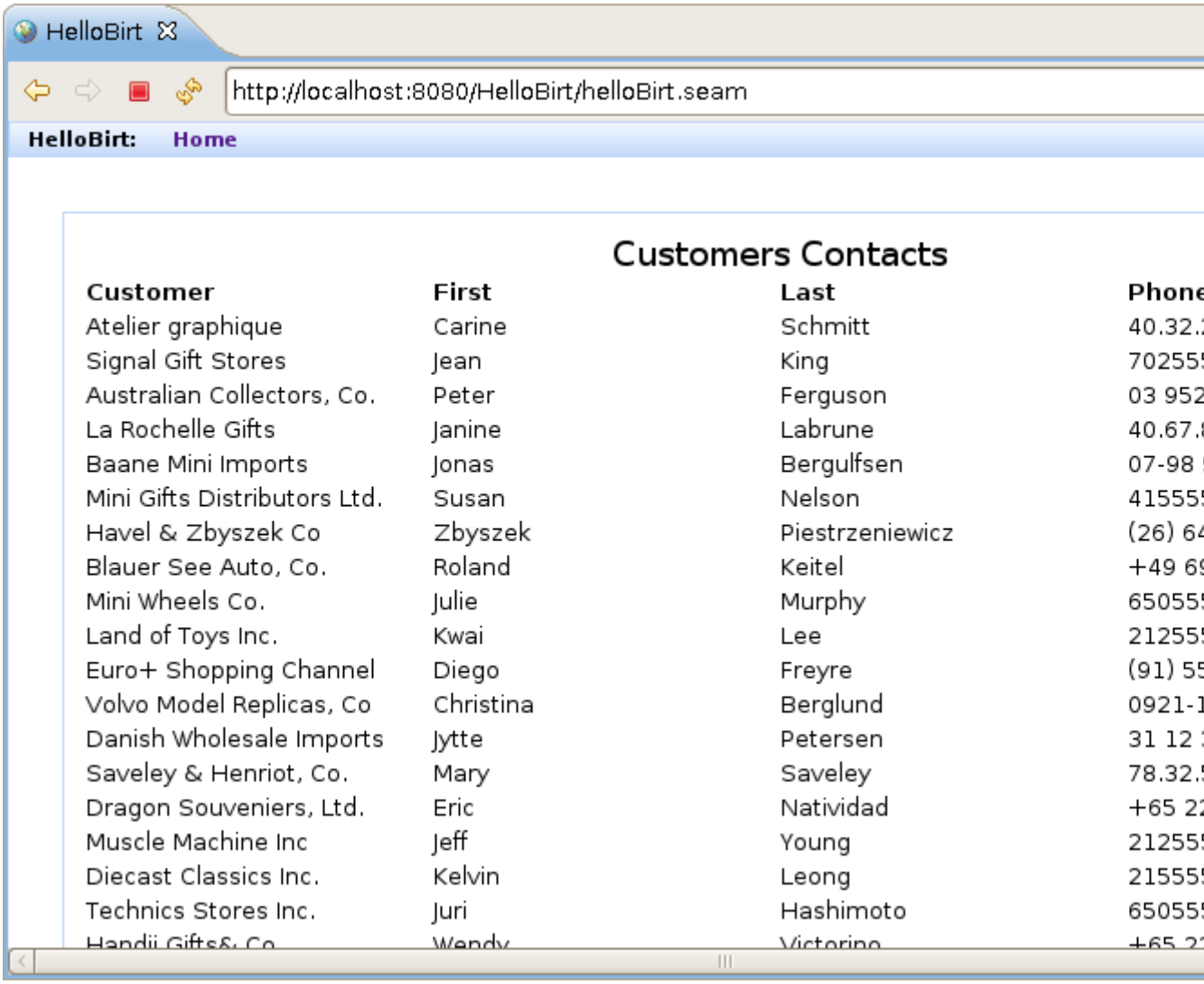

Figure 2.6. Integrating BIRT into Seam Project

2.2. Using Hibernate ODA Data Source

The JBoss BIRT Integration feature includes the Hibernate ODA Data Source which is completely integrated with Hibernate Tools. You can use it the way as you would use any of BIRT ODA drivers.

1. First, you need to reverse engineer from the database to generate Seam entities. You can perform this operation going to *File > New > Seam Generate Entities* in the Seam perspective. More details on the Seam Generate Entities please read *Seam Dev Tools Reference guide* [http://download.jboss.org/jbosstools/nightly-docs/en/seam/html_single/index.html#generate_entities]). In this guide we will use the Employees table of the DATAMODELS database.

Tip:

Before performing Seam Generate Entities, you should have a connection profile adjusted and connected to a database. How to do this see in the *CRUD Database Application* [http://download.jboss.org/jbosstools/nightly-docs/en/seam/html_single/index.html#d0e2725] chapter of the Seam Dev Tools Reference guide.

If you followed the steps described in the *Adding BIRT Functionality to Standard Seam Web Project* chapter, you would have a connection profile already configured.

2. Next you should create a new BIRT report file (*>File > New > Other > Business Intelligence and Reporting Tools > Report*) to represent the data from the Employees table. Call the file employees.rptdesign.

3. Now switch to the BIRT Report Design perspective.

4. In the Data Explorer view right-click the *Data Source* node and choose *New Data Source.*

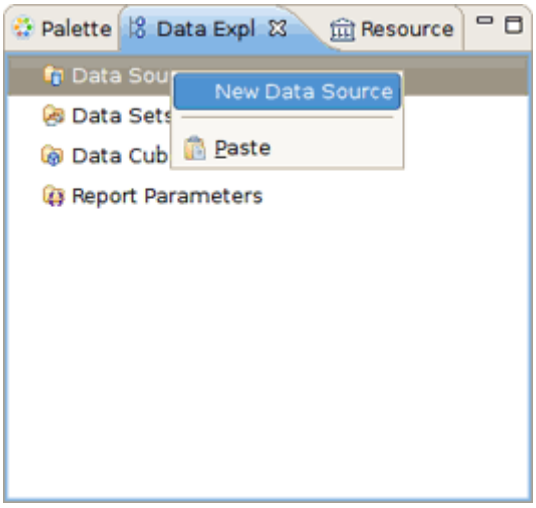

Figure 2.7. Creating a New Data Source

5. The wizard will prompt you to select data source type. Choose *Hibernate Data Source* and give it a meaningful name, for instance *HibernateDataSource* . Hit *Next to proceed.*

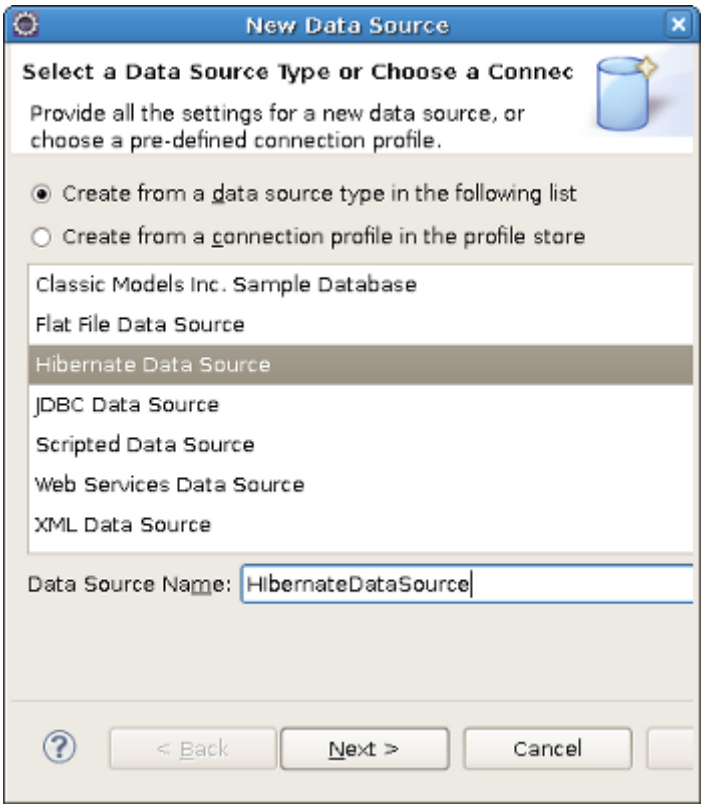

Figure 2.8. Creating Hibernate Data Source

6. On the next wizard's dialog you can leave the everything with default values, press the *Test Connection* button to verify that the connection is established successfully.

   The Hibernate Data Source enables you to specify a Hibernate Configuration or JNDI URL.

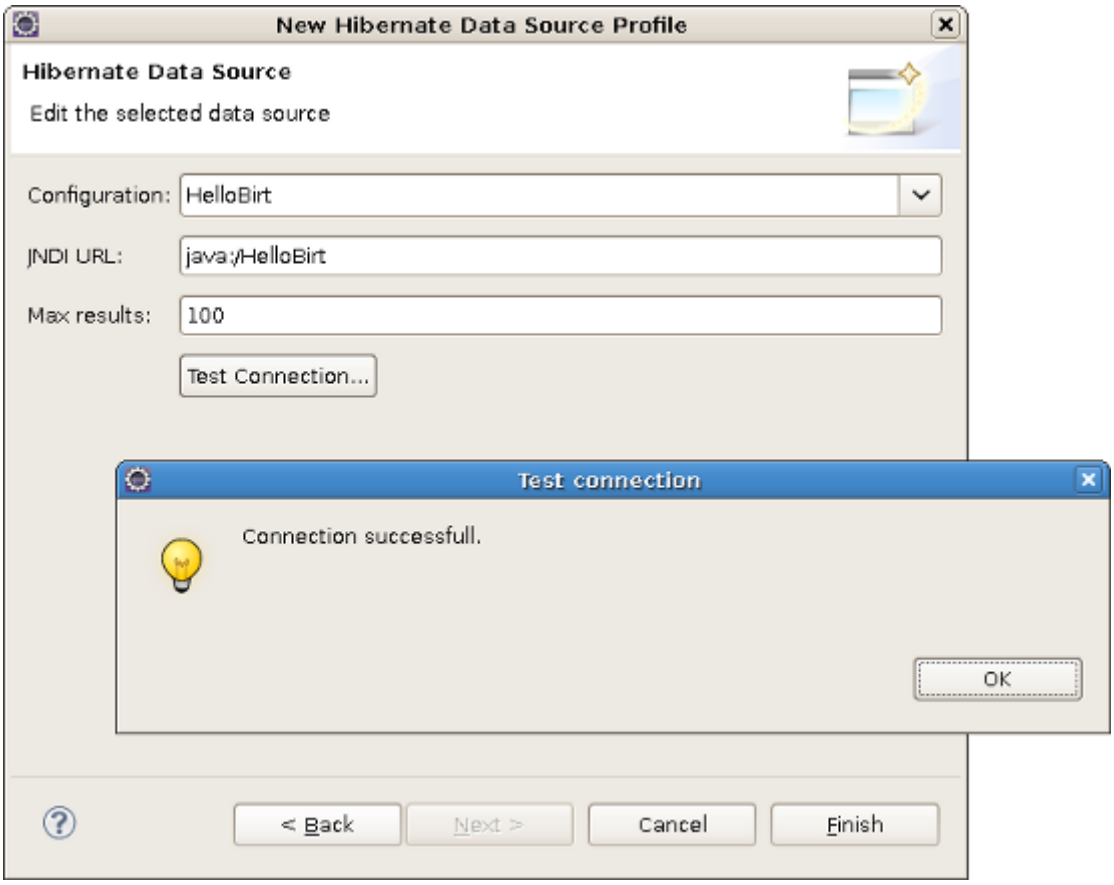

Figure 2.9. Hibernate Data Source Profile

7. Press *Finish* to complete New Data Source wizard.

8. Now you need to configure a new Hibernate ODA data set. Launch the New Data Set wizard. In the Data Explorer View right-click the *Data Set* node and select *New Data Set.*

9. Select HibernateDataSource as target data source and type in the new data set name. Call it HibernateDataSet.

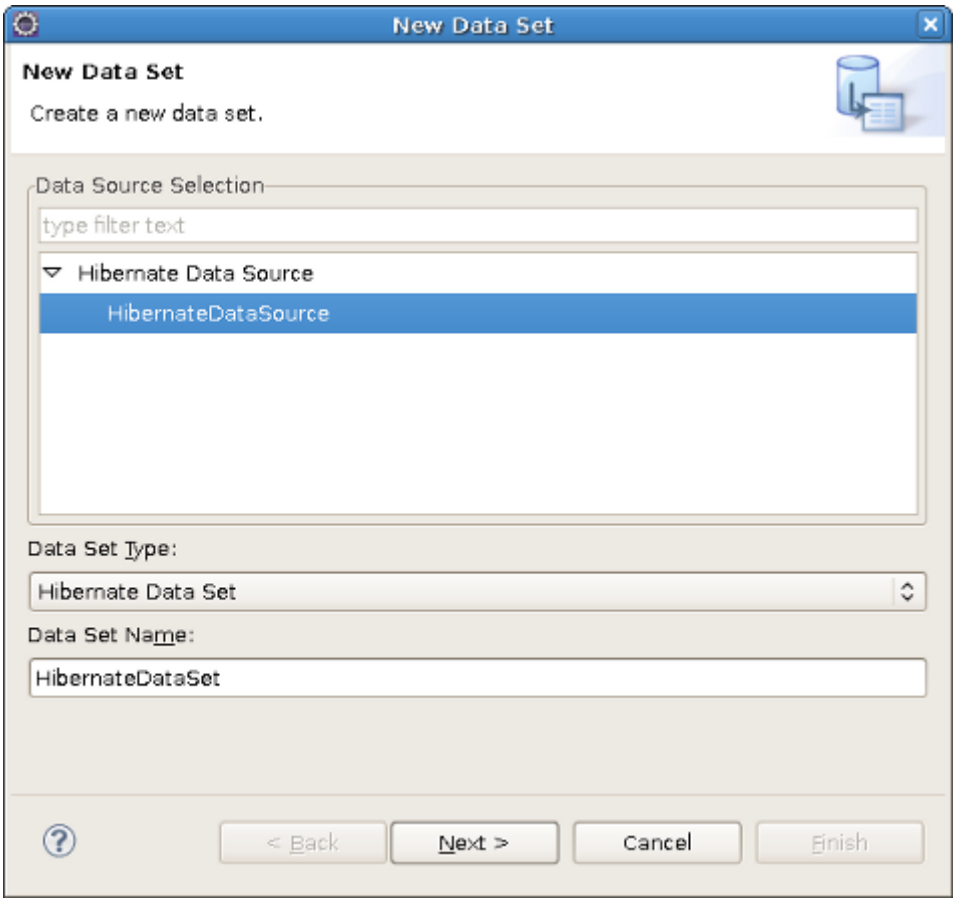

Figure 2.10. Creating a Hibernate ODA Data Set

10. The next dialog of the wizard will help you compose a query for the new data set.

    We will make a report that will print all employees in the database who has Sales Rep job title.

```
...
select jobtitle, firstname, lastname, email
from Employees as employees where employees.jobtitle = 'Sales Rep'
...
```

    To validate the entered query you can press the *Test query* button. All the HQL features like syntax highlighting, content assist, formatting, drag-and-drop, etc., are available to facilitate query composing.

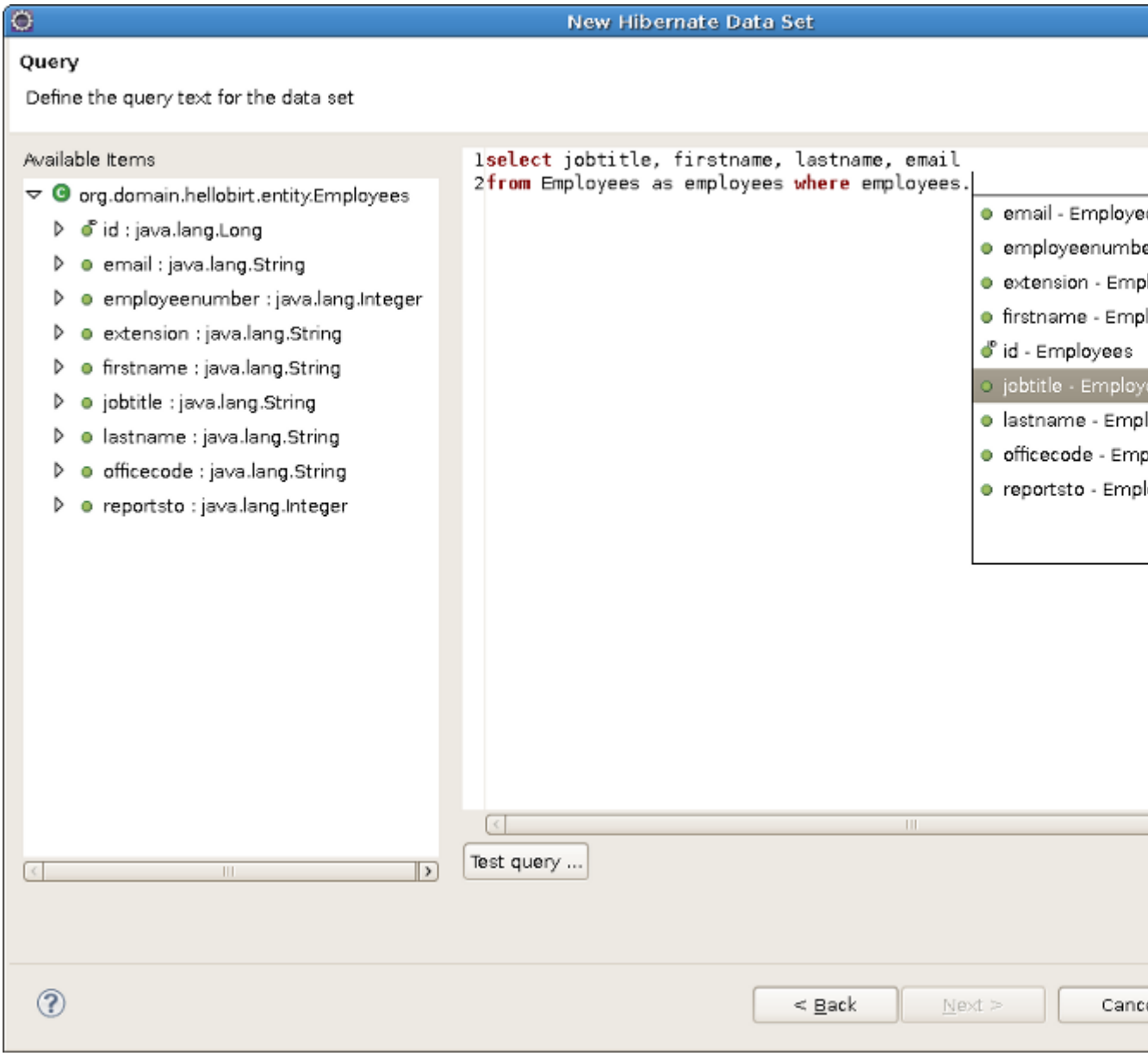

Figure 2.11. Composing query for Data Set

11 Pressing Finish will call the Edit Data Set dialog where you can adjust the parameters of the data set and preview the resulted set. If everything looks good, hit Ok to generate a new data set.

12 Now you can insert the data set items of HibernateDataSet into the employees.rptdesign.

Tip:

If you don't know how to do this we suggest that you refer to the *Eclipse BIRT Tutorial* [http://www.eclipse.org/birt/phoenix/tutorial/].

You can also use parameters in the query to add dynamics to your report. In the previous example we hardcoded the selection criterion in the `where` clause.

To specify the job title on-the-fly your query should look like this:

```
select jobtitle,firstname, lastname,email
from Employees as employees where employees.jobtitle = ?
```

The question mark represents a data set input parameter, which is not the same as a report parameter. Now you need to define an new report parameter to pass the data to the report, call it `JobTitle`. The dataset parameter can be linked to a report parameter. In the Data Explorer view click the Data Set node to open it and right-click on the data set you created previously(in our case it is HibernateDataSet), choose Edit and navigate to the Parameters section. Declare a new data set parameter, name it `jobtitle` and map it to the already existing `JobTitle` report parameter.

13. You report is ready, you can view it by clicking on the Preview tab of the BIRT Report Designer editor.

    You will be prompted to assign a value to the report parameter. For instance you can enter "Sales Rep".

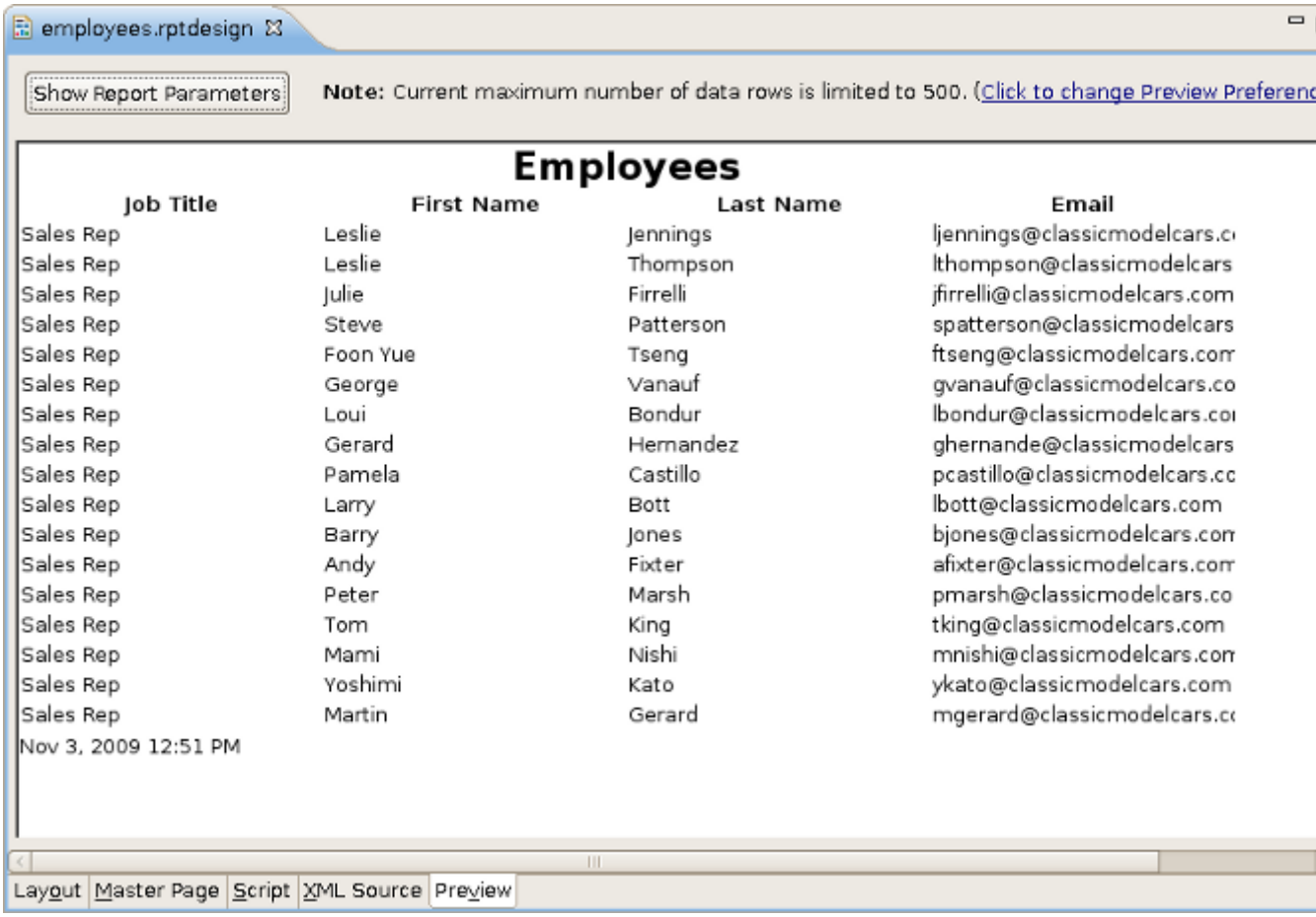

Figure 2.12. Resulted Dynamic Report

2.3. Creating a Dynamic BIRT Report

The *Adding BIRT Functionality to Standard Seam Web Project* and *Using Hibernate ODA Data Sourc* sections tell how to integrate a BIRT report into a Seam web project and how to use Hibernate data source to generate a dynamic report. In this section we will create a Seam web project that can make a dynamic report using the parameters that are defined on a web page.

We will use the PRODUCTS table of DATAMODELS database for the purpose of this demo project. In the first place, you need to generate Seam entities like we did in the previous chapter (*Hibernate ODA Data Source*). The demo application will generate a report about the company's products, whereas it will allow to specify a product line.

1. The next step is to create a Java class that will store the `sortOrder` variable and its assessors, register the class in faces.config.xml. The variable will be needed to pass dynamic data to the report via report parameters, therefore it has to be of session scope.

2. The report will print the data from the `Products` table, hence you need to create a report file first. You can use either the BIRT JDBC Data Source or Hibernate Data Source data source

to create the data set for this project. If you want to use the latter please read the previous chapter *Hibernate ODA Data Source*.

The data set should have at least the following data set items: product vendor, product name, quantity in stock and buy price. The data is retrieved from the database with this query :

```
SELECT  productvedor,
productname,
quantityinstock,
buyprice
FROM Products as products
```

3. Make a table in the report and put each data set item into a column.

4. As it was stated in the beginning of the chapter the report will be dynamic, therefore you need to declare a report parameter first, let it be `sortOrder` and to add the parameter to the query. BIRT offers rich JavaScript API, so you can modify the query programmatically like this:

```
<xml-property name="queryText"><![CDATA[
SELECT  productvedor,
productname,
quantityinstock,
buyprice
FROM Products as products
]]></xml-property>
    <method name="beforeOpen"><![CDATA[
queryString              =              "              ORDER              BY
 products."+reportContext.getParameterValue("sortOrder")+" "+"DESC";
this.queryText = this.queryText+queryString;
]]></method>
```

5. The report is ready. You can preview it to make sure it works properly.

6. To set the report parameter you should create an .xhtml view page, call it Products.xhtml. On the page you can set the value of the `sortOrder` Java bean variable and press the Generate Report button to open another view page that will display the resulted report.

   The source code of the Products.xhtml should be the following:

```
```

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
 xmlns:s="http://jboss.com/products/seam/taglib"
 xmlns:ui="http://java.sun.com/jsf/facelets"
 xmlns:h="http://java.sun.com/jsf/html"
 xmlns:f="http://java.sun.com/jsf/core"
 xmlns:rich="http://richfaces.org/rich"
 xmlns:a4j="http://richfaces.org/a4j" template="layout/template.xhtml">
 <ui:define name="body">
  <rich:panel>
   <f:facet name="header">BIRT Report Generator</f:facet>
   <a4j:form ajaxSubmit="true" reRender="criterion">
    <table>
     <tr>
      <td>Select sort order criterion:</td>
      <td><h:selectOneMenu onchange="submit()"
       value="#{yourJavaBean.sortOrder}"> <!-- Bind to your Java Bean -->
       <f:selectItem itemValue="buyprice" itemLabel="buy price" />
     <f:selectItem itemValue="quantityinstock" itemLabel="quantity in stock" /
>
      </h:selectOneMenu>
      </td>
     </tr>
    </table>
    </a4j:form>
   <s:button
              view="/#{!empty    reportParameters.order    ?    'Products'    :
 'ProductsReport'}.xhtml" id="generate" value="Generate Report" /> <!-- If the
 sertOrder variable is not set the button won't work -->
  </rich:panel>
 </ui:define>
</ui:composition>
```

The logic of the file is quite simple, when the sort order criterion is select the value of `yourJavaBean.sortOrder` is set automatically via Ajax and the report is ready to be generated.

7. Now you need to create the web page that will print the report, name the file ProductsReport.xhtml. The file to output the report should have the following content:

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
 xmlns:s="http://jboss.com/products/seam/taglib"
 xmlns:ui="http://java.sun.com/jsf/facelets"
 xmlns:f="http://java.sun.com/jsf/core"
 xmlns:h="http://java.sun.com/jsf/html"
```

```
 xmlns:b="http://jboss.com/products/seam/birt"
 xmlns:rich="http://richfaces.org/rich" template="layout/template.xhtml">
 <ui:define name="body">
  <rich:panel>
   <f:facet name="header">Products Report</f:facet>
   <b:birt designType="embed" designName="products.rptdesign"
    embeddable="true" masterpage="true">
    <b:param name="sortOrder" value="#{yourJavaBean.sortOrder}" />
   </b:birt>
  </rich:panel>
</ui:define>
</ui:composition>
```

As you know from the *Adding BIRT Functionality to Standard Seam Web Project* chapter, before using the BIRT Integration framework tags on the page you need to declare the tag library and specify the name space with this line:

```
 xmlns:b="http://jboss.com/products/seam/birt"
```

The dynamics to the report adds this line:

```
<b:param name="sortOrder" value="#{yourJavaBean.sortOrder}" />
```

We bound the `sortOrder` report parameter to Java Bean variable `value="#{yourJavaBean.sortOrder}"` using EL expression, and the value to the variable is assigned in the Products.xhtml file.

By default if you embed a report into HTML page the HTML-format report contains the <html>, <head>, <body> etc., tags. However if your HTML page already has those tags, you can rid of them using the `embeddable="true"` attribute of the **<b:birt>** component.

8. Deploy the project onto the server and open your browser to see the report is successfully generated. You should navigate to `http://localhost:8080/yourProjectName/Products.seam` to select the criterion and press the Generate Report button. You will be redirected to the `http://localhost:8080/HelloBirt/ProductsReport.seam`

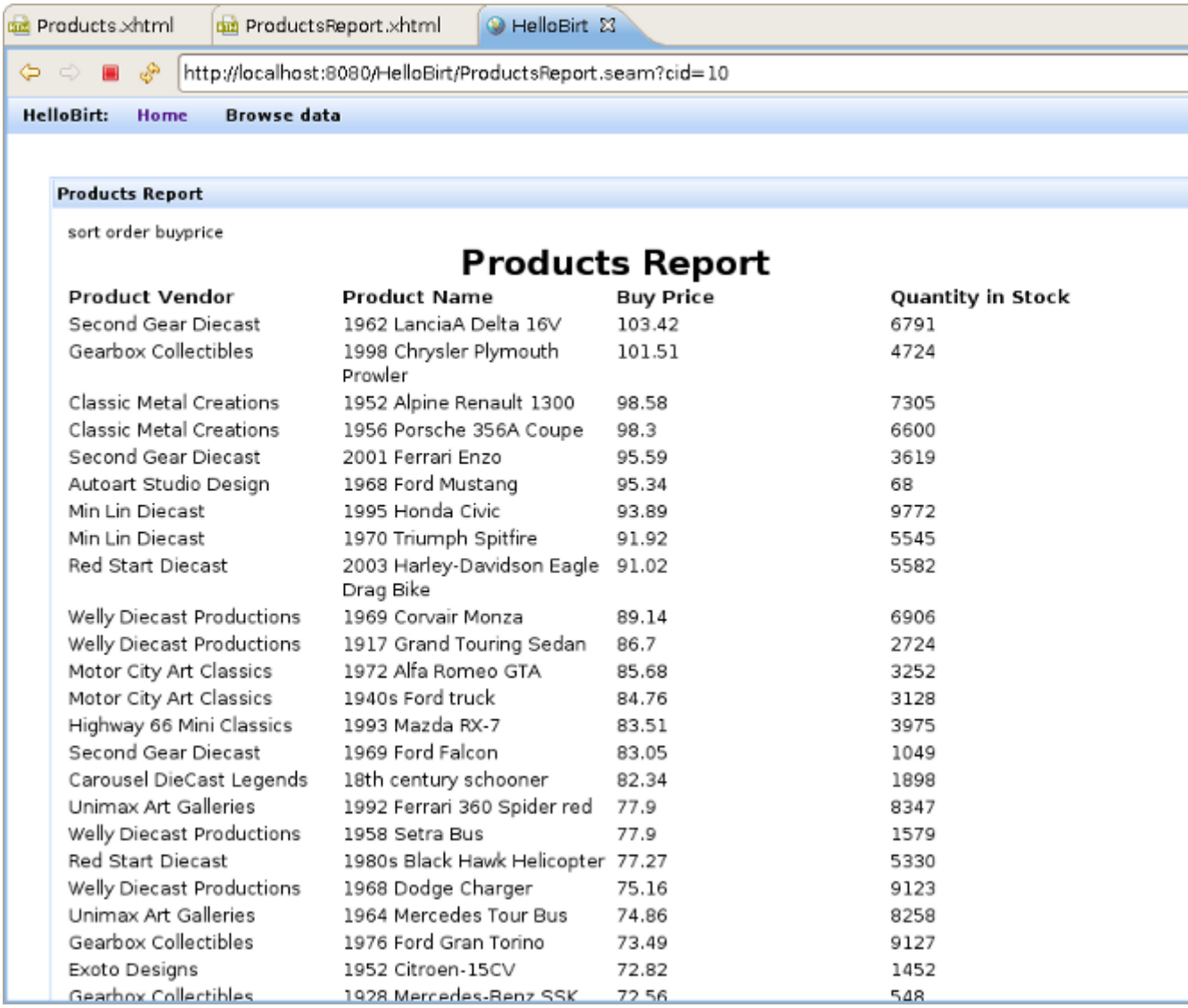

Figure 2.13. Dynamic Report

Thus, a Seam project that includes the BIRT facet can be deployed as any project. If you define the Hibernate ODA driver, the JBoss BIRT engine will use JNDI URL that has to be bound to either Hibernate Session Factory or Hibernate Entity Manager Factory. If you don't specify the JNDI URL property, our engine will try the following JNDI URLs:

- *java:/<project_name>*
- *java:/<project_name>EntityManagerFactory*

When creating a Seam EAR project, Hibernate Entity Manager Factory is bound to *java:/{projectName}EntityManagerFactory*. All you need to do is to use the Hibernate Configuration created automatically. You can use default values for the Hibernate Configuration and JNDI URL within the BIRT Hibernate Data Source.

When using a Seam WAR project, neither HSF nor HEMF aren't bound to JNDI by default. You have to do this manually. For instance, HSF can be bound to JNDI by adding the following property to the *persistence.xml* file:

```
<property name="hibernate.session_factory_name" value="java:/projectname"/>
```

And you can use *java:/projectname* as the JNDI URL property when creating a BIRT Hibernate Data Source.

Note:

If you want to test this feature using PDE Runtime, you need to add *osgi.dev=bin* to the *WebContent/WEB-INF/platform/configuration/config.ini* file.

In conclusion, the main goal of this document is to get you to know with a full feature set that JBoss BIRT Tools provide. Thus if you have some questions, comments or suggestions on the topic, please feel free to ask in the *JBoss Tools Forum* [http://www.jboss.org/index.html?module=bb&op=viewforum&f=201]. You can also influence on how you want to see JBoss Tools docs in future leaving your vote on the article *Overview of the improvements required by JBossTools/JBDS Docs users* [http://www.jboss.org/community/docs/DOC-10795].

JBoss BIRT Integraion Framework API Reference

3.1. <b:birt> Component

The **<b:birt>** component servers to integrate a BIRT report into Seam/JSF container. The **<b:birt>** tag recognizes most of the parameters described on the BIRT *Report Viewer Parameters* [http://www.eclipse.org/birt/phoenix/deploy/viewerUsage2.2.php#parameters] page, though it has attributes of its own.

Table 3.1. <b:birt> Component Reference

Attribute	Correspondance to BIRT Report Viewer parameters	Description/
designType	-	Corresponds to the BIRT servlet mappings. Possible values are run, frameset, preview and embed. If the attribute is set to embed the component is used for embeddable html. This type ignores all the attributes except designName and masterpage.
embeddable	-	A BIRT report occupies the whole page by default. It contains the <html>, <head>, <body> etc., tags. Embedded report can be a part of another page. It doesn't contain the above mentioned tags. Only HTML report can be embedded.
designName	__report	Sets the name of the report design to process. This can be an absolute path or relative to the working folder. Valid values are run, frameset and preview

Attribute	Correspondance to BIRT Report Viewer parameters	Description/
format	__format	Specifies the desired output format, such as pdf, html, doc, ppt, or xls.
title	__title	Sets the report title.
showtitle	__showtitle	Determines if the report title is shown in the frameset viewer. Defaults to true.
toolbar	__toolbar	Determines if the report toolbar is shown in the frameset viewer.Defaults to true. Valid values are true and false.
navigationbar	__navigationbar	Determines if the navigation bar is shown in the frameset viewer. Defaults to true. Valid values are true and false.
document	__document	Sets the name for the rptdocument. The document is created when the report engine separates run and render tasks, and is used to support features like table of contents and pagination. This setting can be an absolute path or relative to the working folder. If no document parameter is used, a unique document is created in the document folder.
locale	__locale	Specifies the locale for the specific operation. Note that this will override the default locale.
svg	__svg	Specifies whether SVG is supported.
page	__page	Specifies specific page to render.
pagerange	__pagerange	Specifies page range to render. Eg 1-4,7.

Attribute	Correspondance to BIRT Report Viewer parameters	Description/
masterpage	__masterpage	Indicates that the report master page should be used or not. Validvalues are true and false.
overwrite	__overwrite	This setting if set to true will force an overwrite of the existing report document.
bookmark	__bookmark	Specifies a specific bookmark within the report to load. The viewer will automatically load the appropriate page.
rtl	__rtl	Specifies whether to display the report in right to left format. This setting defaults to false.
fittopage	__fittopage	Specifies whether PDF generation should fit content to a page. Valid values are true and false.
resourceFolder	__resourceFolder	Specifies the resource folder to use. This setting will override the default setting in the web.xml. The resource folder is used to locate libraries, images, and resource files.

3.2. <b:param> Component

The **<b:param>** tag describes report parameter.

Table 3.2. <b:param> Component

Attribute	Description
name	The parameter name. Required.
value	The parameter value. Required.
isnull	Specifies that a report parameter has a null value
islocale	Specifies whether the parameter is localized

Other Relevant Resources on the Topic

All JBoss Developer Studio/JBoss Tools release documentation you can find at *http://docs.jboss.org/tools* [http://docs.jboss.org/tools/] in the corresponding release directory.

The latest documentation builds are available at *http://download.jboss.org/jbosstools/nightly-docs* [http://download.jboss.org/jbosstools/nightly-docs/].