JBoss BIRT Integration User Guide

Version: 3.3.0.GA

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.

Installation Instructions

2.1. Installing JBoss Tools Plugins

The plugin can be installed in Eclipse from the JBoss.org update site. JBoss Tools 3.2 requires Eclipse 3.6, which can be downloaded from the *Eclipse web site* [http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/].

To install the JBoss Tools plugins start Eclipse and select **Help** → **Install New Software...**

Figure 2.1. Install New Software

Click the **Add...** button.

Figure 2.2. Install Dialog

This will display the **Add Repository** dialog. Enter **JBoss.org Tools** in the **Name** field, and *http://download.jboss.org/jbosstools/updates/JBossTools-3.2.0.GA/* in the **Location** field. Click the **OK** button to save the changes and close the dialog.

Figure 2.3. Add Repository

The **JBoss.org Tools** site will be selected in the **Work with** drop down list, and after a moment the list of plugins that are included in the JBoss Tools package will be listed. From this list you can individually select the plugin, or select the **All JBoss Tools 3.2.0** option to install all the plugins.

Figure 2.4. Available Software

Click the **Next** button to calculate the system requirements and dependencies (this may take a little while). You will then be given an opportunity to review the plugins that will be installed.

Figure 2.5. Installation review

Click the **Next** button to install the selected plugins. You will be prompted to accept the various license agreements that cover the plugins that are to be installed. Review the licenses, select the **I accept the terms of the license agreements** option, and click the **Finish** button to install the plugins.

Figure 2.6. License review

Wait while the plugins are downloaded and installed.

Figure 2.7. Installing Software

You may be prompted with a warning informing you that you are attempting to install unsigned content. Click the **OK** button to continue.

Figure 2.8. Unsigned Software Warning

You will then have to restart Eclipse to apply the new plugins. Click the **Restart Now** button to restart Eclipse.

Figure 2.9. Restart Eclipse

When Eclipse is first started with the JBoss Tools plugins installed, you may be prompted to allow or disallow anonymous statistics to be sent to the JBoss development team. You can find more information on the data that is sent in the *Getting Started Guide* [http://docs.redhat.com/docs/en-US/JBoss_Developer_Studio/4.0/html-single/Getting_Started_Guide/index.html#Collected_usage_information_guide]. Click the **Yes** to allow the statistics to be sent, or click the **No** button if you prefer not to send this data.

Figure 2.10. JBoss Tools Usage

The plugin is now installed and ready to use.

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.

3.1. Adding BIRT Functionality to Standard Seam Web Project

This section discusses the process of integrating BIRT into a Seam web project.

To follow this guide you will need to have the 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:**
>
> *JBoss Seam 2.2.1 GA* [http://sourceforge.net/projects/jboss/files/JBoss %20Seam/2.2.1.Final/] and *JBoss Application Server 5.1.0 GA* [http:// sourceforge.net/projects/jboss/files/JBoss/JBoss-5.1.0.GA/] were used in the examples presented in this guide.

3.1.1. Creating Seam Web Project with Birt Facet

It is recommended that you open the Seam Perspective by selecting **Window** → **Open Perspective** → **Other** → **Seam**. This perspective provides convenient access to all the Seam tools.

To create a new Seam Web project select **File** → **New** → **Seam Web Project**. If the Seam Perspective is not active, select **File** → **New** → **Other** → **Seam** → **Seam Web Project**.

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

Figure 3.1. Creating Seam Web Project

2. In the **Configuration** group select the Seam framework version 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.
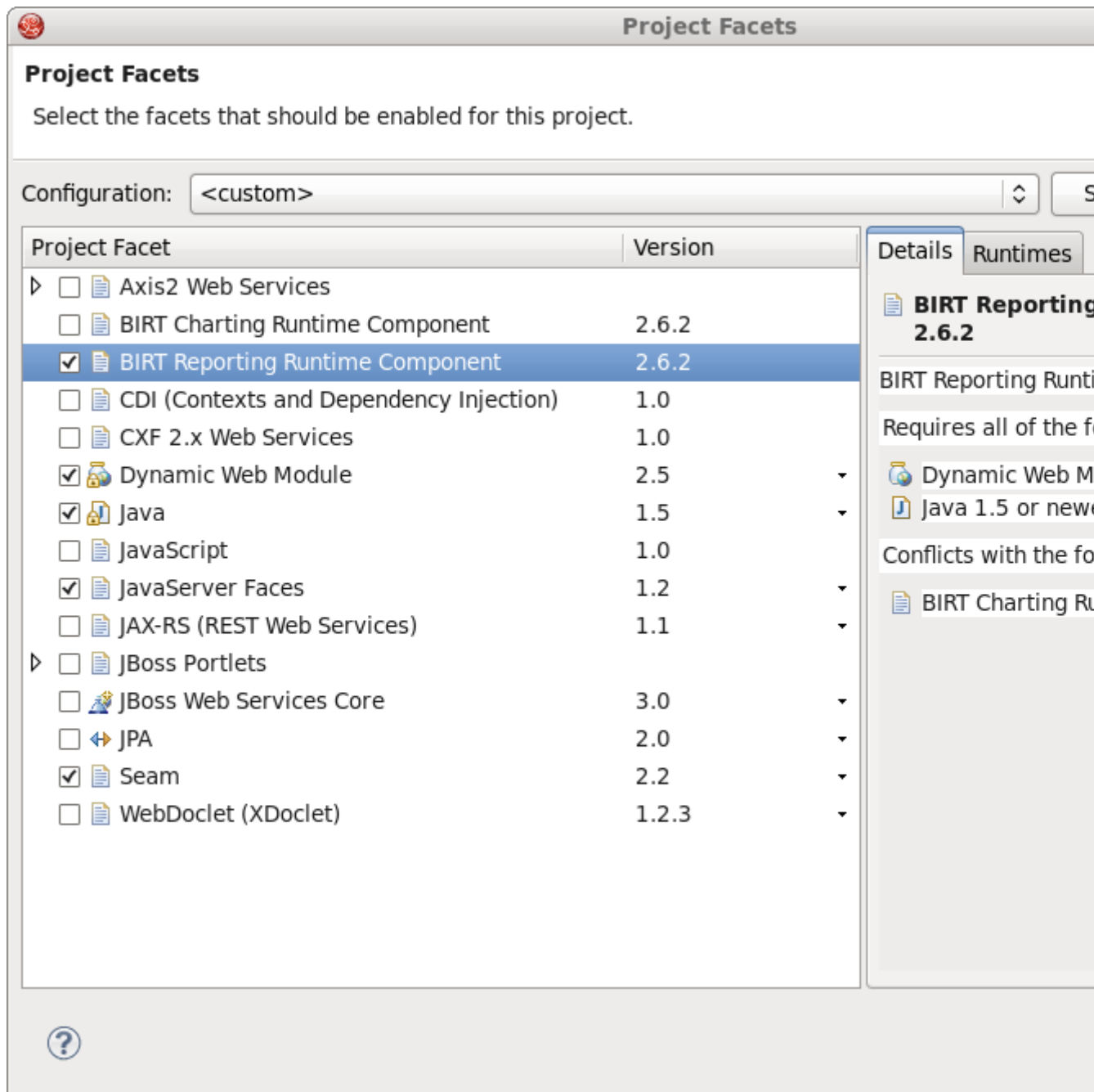
Figure 3.2. Adding the Birt Reporting Runtime Component Facet

Alternatively you can select the **JBoss BIRT Integration Web Project** configuration option from the drop-down list in the **Configuration** group.
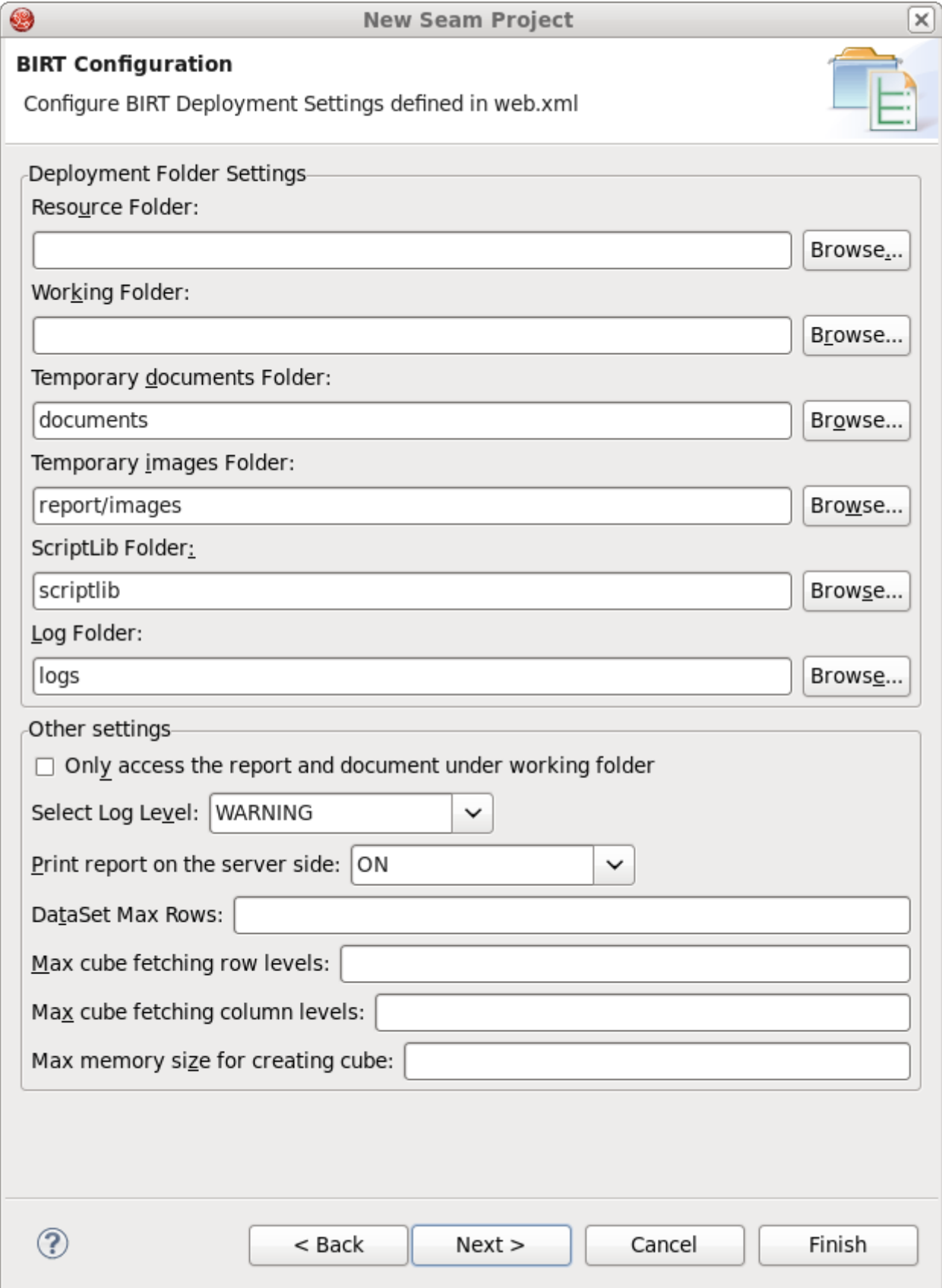
Figure 3.3. Choosing the JBoss BIRT Integration Web Project Configuration

4. You may leave the next two pages with default values; just click the **Next** button to proceed.

5. On the **Birt Configuration** page you can modify the BIRT deployment settings. These settings can also be edited afterwards in the `web.xml` file included in the generated project. Keep the default values for now.

6. You can also leave the default options on the **JSF Capabilities** page.

7. On the **Seam Facet** page you should specify the Seam runtime and Connection profile. Please note that the Seam runtime must be the same version you initially specified in the project settings (See *Figure 3.1, "Creating Seam Web Project"*).

   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/3.2.0.GA/en/seam/html_single/index.html#seamFacet] chapter of Seam Dev Tools Reference Guide.

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

3.1.2. 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` in the `WebContent` folder.

   The report should print the data from the `CLASSICMODELS.CUSTOMERS` table of the **BIRT Classic Models Sample Database**, namely:

   a. Customer number (`CLASSICMODELS.CUSTOMERS.CUSTOMERNAME`)

   b. Contact person first name (`CLASSICMODELS.CUSTOMERS.CONTACTFIRSTNAME`)

   c. Contact person last name (`CLASSICMODELS.CUSTOMERS.CONTACTLASTNAME`)

   d. 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, the steps required to make the report will not be shown. For more information on creating a BIRT report file please read the *BIRT documentation* [http://eclipse.org/birt/phoenix/tutorial/].

   > **i** **Tip**
   >
   > You can download the `helloBirt.rptdesign` file *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` folder 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>
```
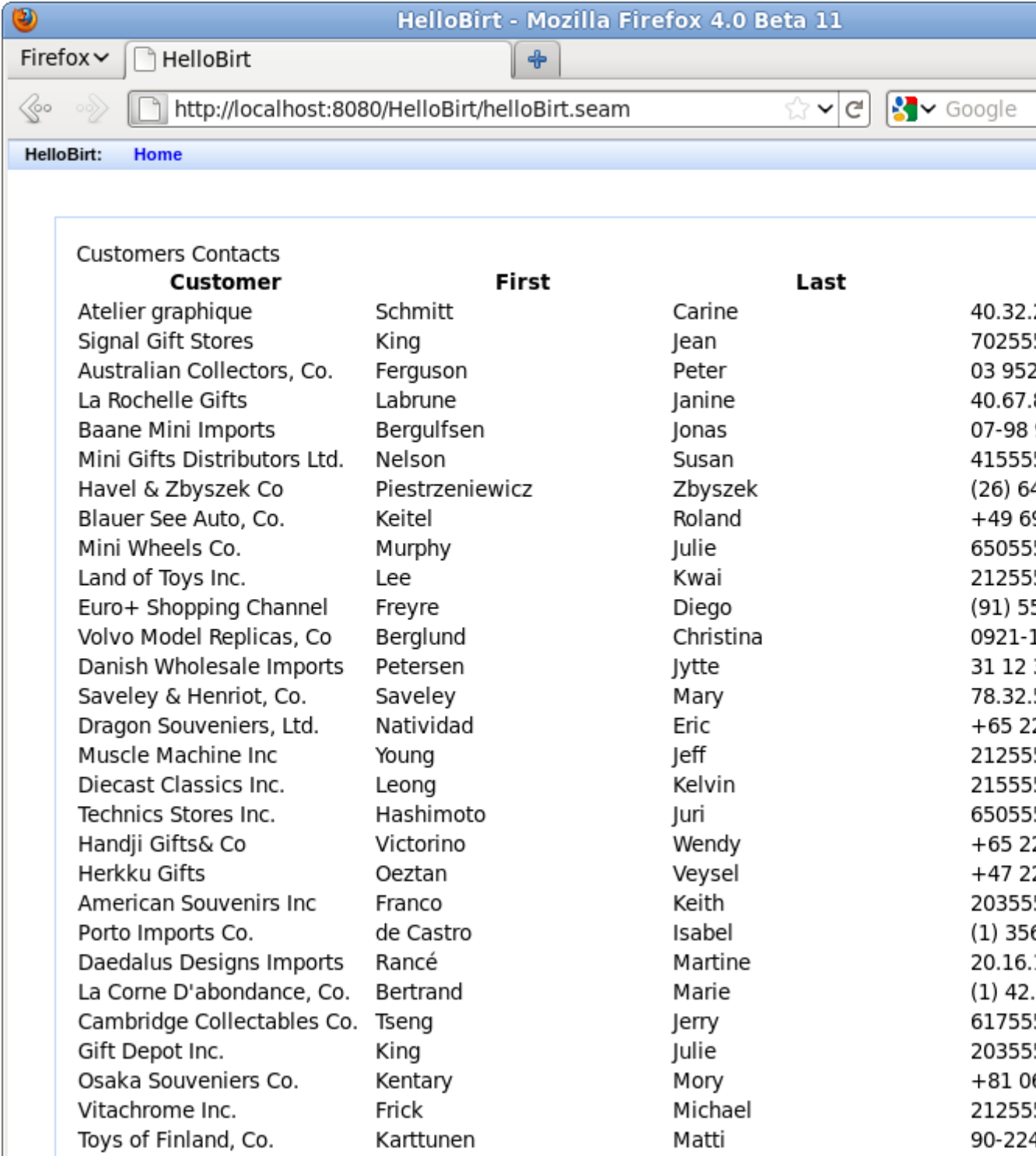
From this listing above you see that the title of the report is set via `<b:param>` by setting the parameter name and defining the `value` attribute with the `Customers Contacts` value.

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 select **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 favorite browser and point it to *http://localhost:8080/ HelloBirt/helloBirt.seam*.

Figure 3.6. Integrating BIRT into Seam Project

3.2. 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 Developers 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, which can be downloaded from the *Getting Started Guide* [http://docs.redhat.com/docs/en-US/JBoss_Developer_Studio/4.0/html-single/Getting_Started_Guide/index.html#create_seam_application].

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

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`, and save it in the `WebContent` folder.

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**.
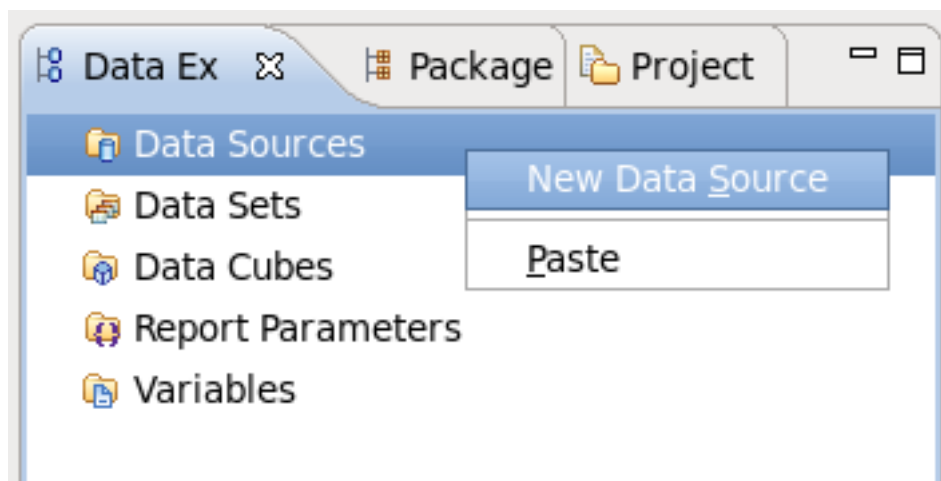
Figure 3.7. Creating a New Data Source

5. The wizard will prompt you to select data source type. Choose **Hibernate Data Source** and give it a meaningful name, for instance **HibernateDataSource**. Click the **Next** button to proceed.
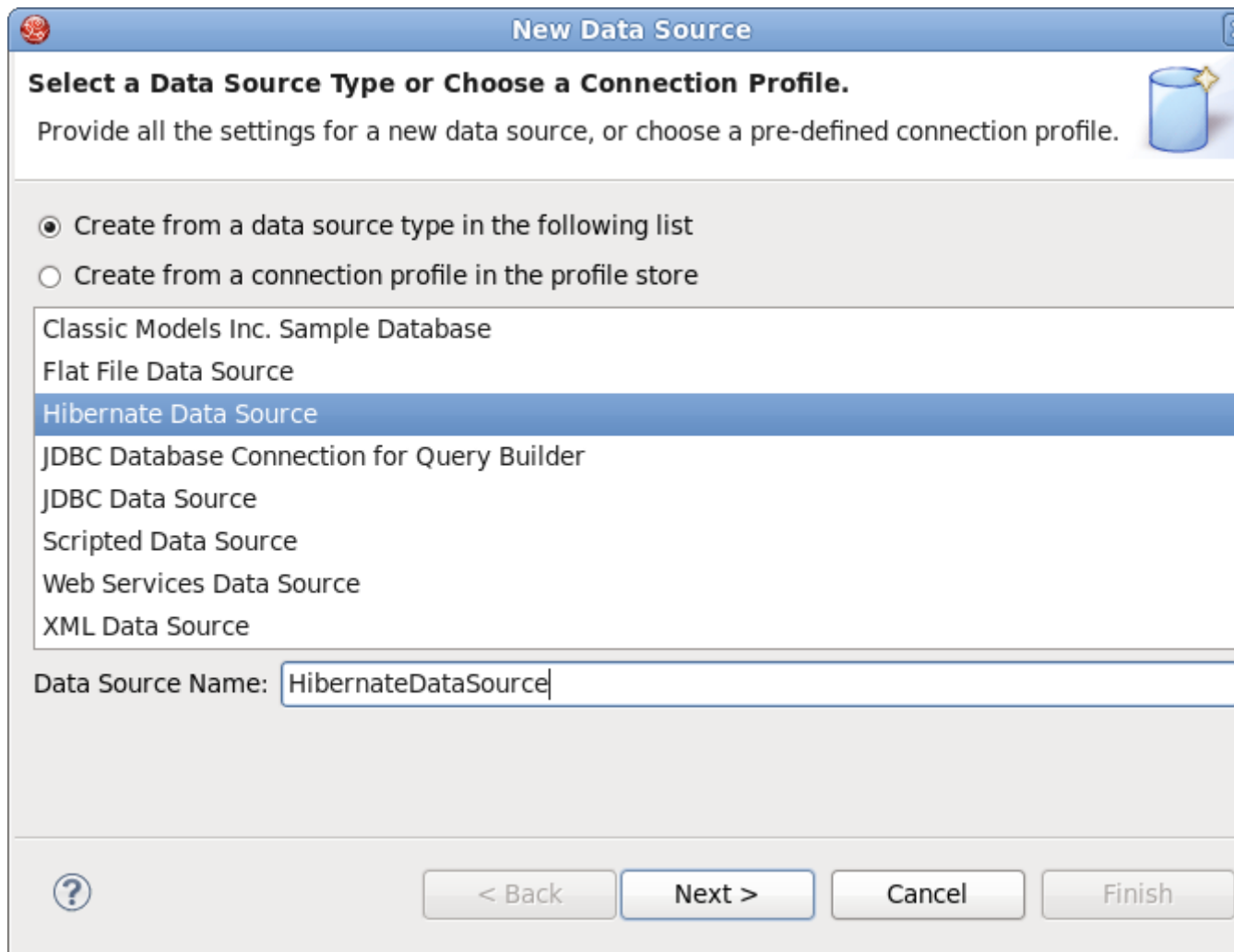
Figure 3.8. Creating Hibernate Data Source

6. On the next wizard's dialog you can leave the everything with default values, click 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.
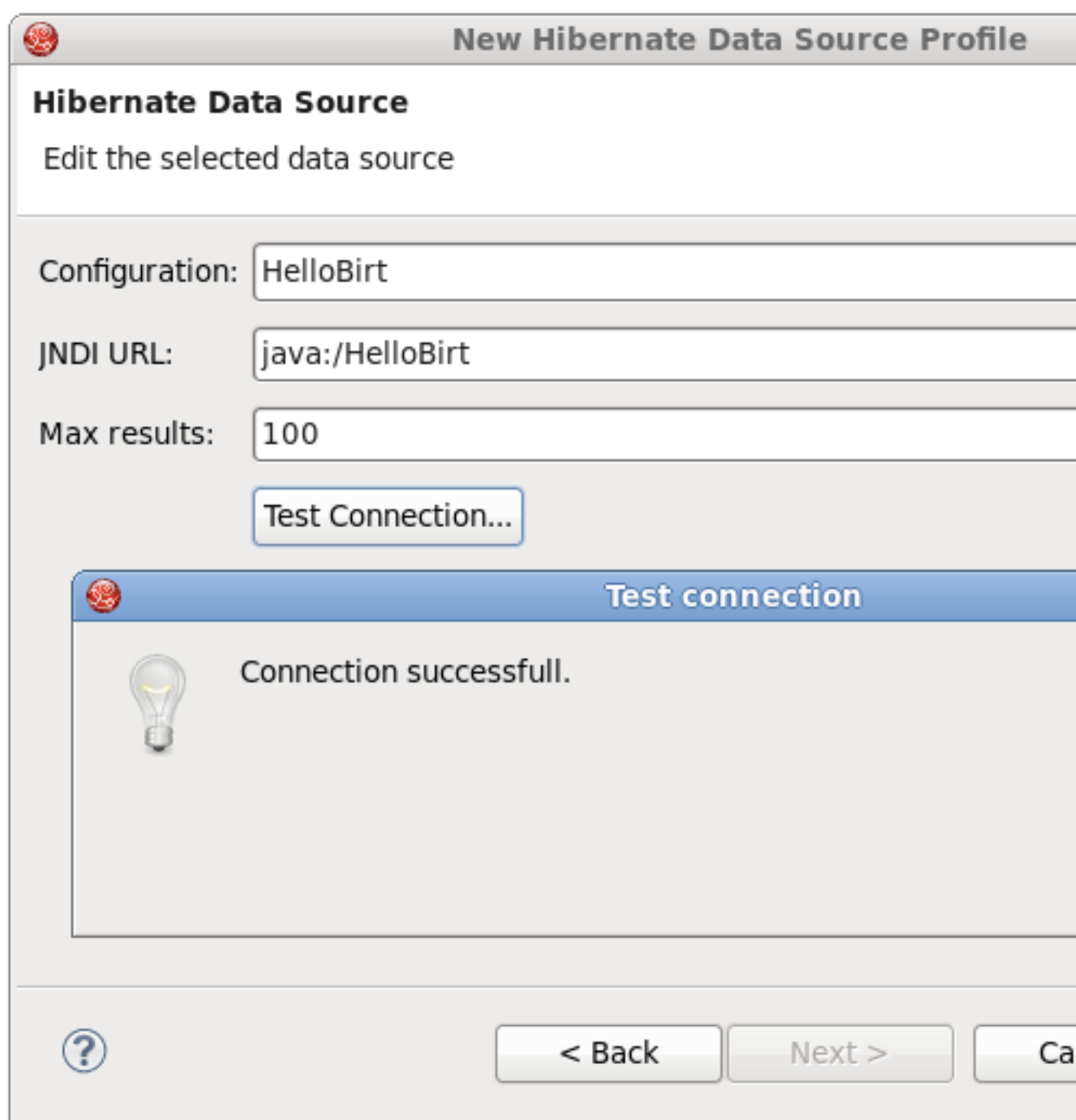
Figure 3.9. Hibernate Data Source Profile

7. Click the **Finish** button 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**.
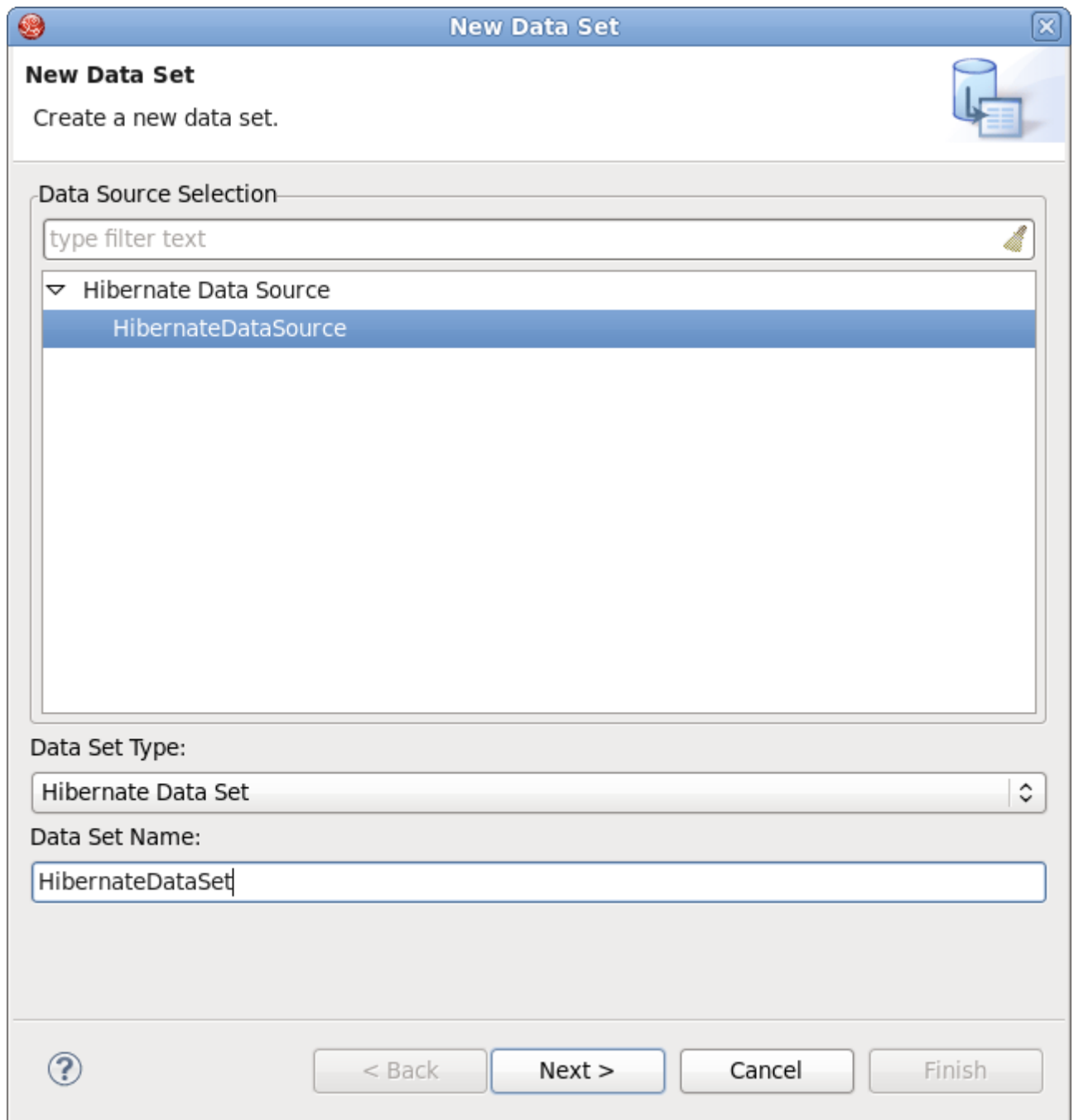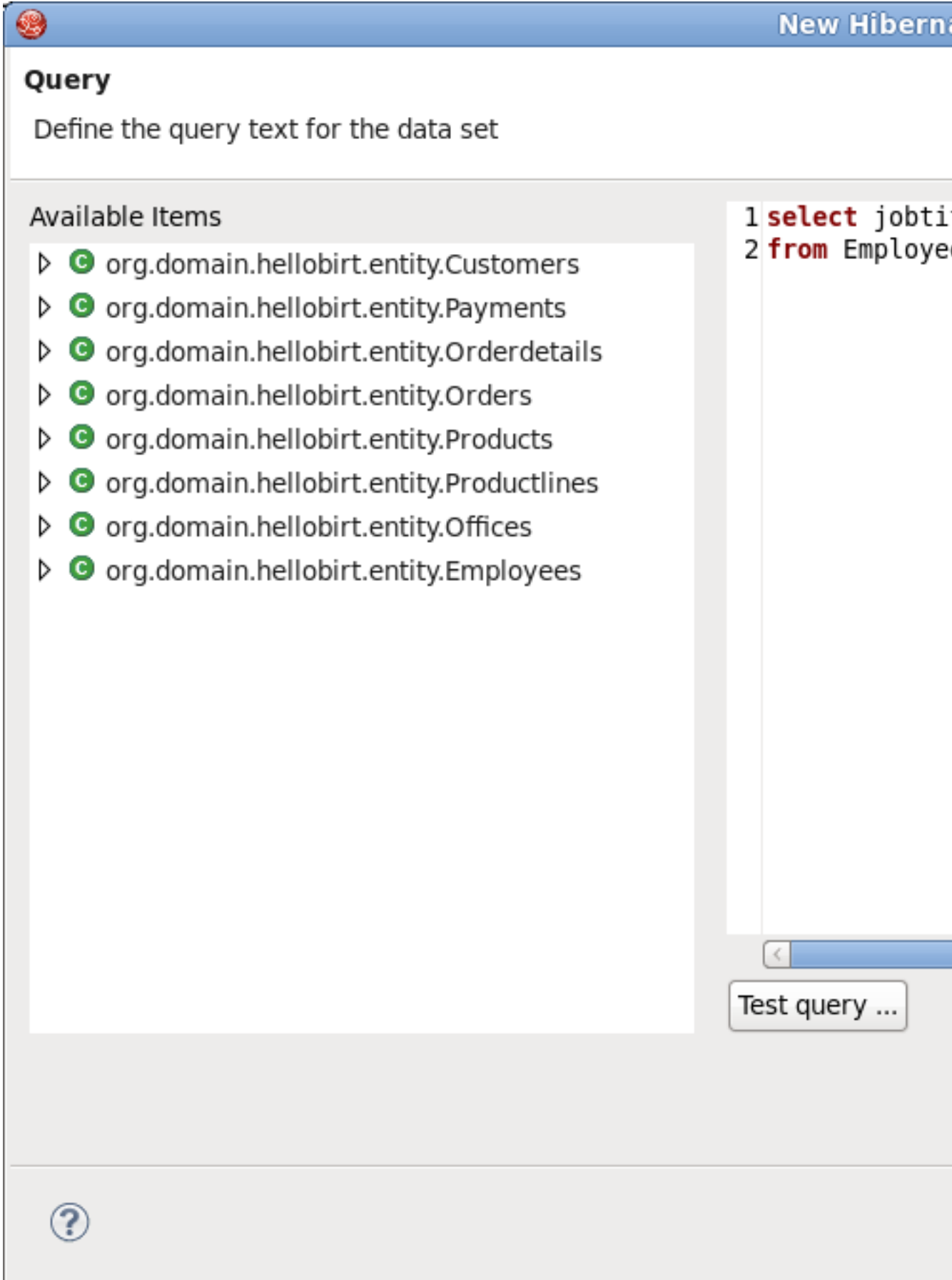
Figure 3.10. Creating a Hibernate ODA Data Set

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.

Query

Define the query text for the data set

Available Items

▷ Ⓒ org.domain.hellobirt.entity.Customers
▷ Ⓒ org.domain.hellobirt.entity.Payments
▷ Ⓒ org.domain.hellobirt.entity.Orderdetails
▷ Ⓒ org.domain.hellobirt.entity.Orders
▷ Ⓒ org.domain.hellobirt.entity.Products
▷ Ⓒ org.domain.hellobirt.entity.Productlines
▷ Ⓒ org.domain.hellobirt.entity.Offices
▷ Ⓒ org.domain.hellobirt.entity.Employees

```
1 select jobti
2 from Employe
```

Test query ...

11.Clicking the **Finish** button 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, click the **OK** button to generate a new data set.

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

> **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 hard coded 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".

🔳 employees.rptdesign ✕

Refresh Report   **Note:** Current maximum number of data rows is limit

Job Title	First Name	Last Name	
Sales Rep	Leslie	Jennings	ljer
Sales Rep	Leslie	Thompson	lth
Sales Rep	Julie	Firrelli	jfir
Sales Rep	Steve	Patterson	spa
Sales Rep	Foon Yue	Tseng	fts
Sales Rep	George	Vanauf	gva
Sales Rep	Loui	Bondur	lbo
Sales Rep	Gerard	Hernandez	ghe
Sales Rep	Pamela	Castillo	pca
Sales Rep	Larry	Bott	lbo
Sales Rep	Barry	Jones	bjo
Sales Rep	Andy	Fixter	afi
Sales Rep	Peter	Marsh	pm
Sales Rep	Tom	King	tki
Sales Rep	Mami	Nishi	mn
Sales Rep	Yoshimi	Kato	yka
Sales Rep	Martin	Gerard	mg

Mar 9, 2011 7:10 AM

Layout | Master Page | Script | XML Source | Preview

3.3. Creating a Dynamic BIRT Report

Section 3.1, "Adding BIRT Functionality to Standard Seam Web Project" and *Section 3.2, "Using Hibernate ODA Data Source"* describe how to integrate a BIRT report into a Seam web project and how to use a 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 *Classic Models Inc. Sample Database* for the purpose of this demo project. The demo application will generate a report about the company's products, and allow the user to specify how the report will be sorted. To begin with, we need to generate Seam entities like we did in the previous *Section 3.1, "Adding BIRT Functionality to Standard Seam Web Project"*.

1. The next step is to create a Java class that will store the `sortOrder` variable and its assessors. The variable will be required to pass dynamic data to the report via report parameters; therefore it has to be of session scope.

   The code below shows a simple JavaBean class called `ReportJB`.

```java
import java.io.Serializable;

import org.jboss.seam.ScopeType;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.annotations.Scope;

@Name("ReportJB")
@Scope(ScopeType.SESSION)
public class ReportJB implements Serializable
{
 private static final long serialVersionUID = 1L;
 protected String sortOrder = "buyprice";

 public String getSortOrder()
 {
  return sortOrder;
 }

 public void setSortOrder(String value)
 {
  sortOrder = value;
 }

 public ReportJB()
 {
```

```
  }
}
```

2. The report will print the data from the *Products* table. Create a new report file file called `ProductsReport.rptdesign` in the `WebContent` folder. 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 *Section 3.2, "Using 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 productvendor, productname, quantityinstock, buyprice
FROM CLASSICMODELS.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. Call this parameter `sortOrder` and to add the parameter to the query.

Figure 3.13. Report Parameter

BIRT offers rich JavaScript API, so you can modify the query programmatically like this (the `xml-property` tag shown below should already be present in the report):

```
<xml-property name="queryText"><
![CDATA[
SELECT productvendor, productname, quantityinstock, buyprice
FROM CLASSICMODELS.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 page, call it ProductForm.xhtml, and place it in the WebContent folder. On the page you can set the value of the sortOrder Java bean variable and click the **Generate Report** button to open another view page that will display the resulted report.

   The source code of the ProductForm.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="#{ReportJB.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 selected the value of ReportJB.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="#{ReportJB.sortOrder}" />
   </b:birt>
  </rich:panel>
 </ui:define>
</ui:composition>
```

As you know from *Section 3.1, "Adding BIRT Functionality to Standard Seam Web Project"*, 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"
```

To set the `sortOrder` report parameter add this line:

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

We bound the `sortOrder` report parameter to Java Bean variable `value="#{ReportJB.sortOrder}"` using EL expression, with the `ReportJB.sortOrder` variable having its value assigned in the `ProductsForm.xhtml` file.

> **ℹ Tip**
>
> 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/ProductForm.seam* to select the criterion and press the **Generate Report** button. You will be redirected to *http:// localhost:8080/yourProjectName/ProductsReport.seam*.

| ProductsReport.xhtml | ProductForm.xhtml | ReportJB.java |

http://localhost:8080/HelloBirt/ProductsReport.seam;js

HelloBirt: Home   Browse data

Products Report

PRODUCT VENDOR	PRODUCT NAME	QUAN
Min Lin Diecast	1969 Harley Davidson Ultima	7933
Classic Metal Creations	1952 Alpine Renault 1300	7305
Highway 66 Mini Classics	1996 Moto Guzzi 1100i	6625
Red Start Diecast	2003 Harley-Davidson Eagle	5582
Motor City Art Classics	1972 Alfa Romeo GTA	3252
Second Gear Diecast	1962 LanciaA Delta 16V	6791
Autoart Studio Design	1968 Ford Mustang	68
Second Gear Diecast	2001 Ferrari Enzo	3619
Welly Diecast Productions	1958 Setra Bus	1579
Unimax Art Galleries	2002 Suzuki XREO	9997
Welly Diecast Productions	1969 Corvair Monza	6906
Welly Diecast Productions	1968 Dodge Charger	9123
Second Gear Diecast	1969 Ford Falcon	1049
Studio M Art Models	1970 Plymouth Hemi Cuda	5663
Exoto Designs	1957 Chevy Pickup	6125
Welly Diecast Productions	1969 Dodge Charger	7323
Studio M Art Models	1940 Ford Pickup Truck	2613
Highway 66 Mini Classics	1993 Mazda RX-7	3975
Motor City Art Classics	1937 Lincoln Berline	8693
Studio M Art Models	1936 Mercedes-Benz 500K Sp	8635
Classic Metal Creations	1965 Aston Martin DB5	9042
Red Start Diecast	1980s Black Hawk Helicopter	5330
Welly Diecast Productions	1917 Grand Touring Sedan	2724
Gearbox Collectibles	1948 Porsche 356-A Roadster	8826
Min Lin Diecast	1995 Honda Civic	9772

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 are 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 describe the full feature set that JBoss BIRT Tools provide. If you have any 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

4.1. <b:birt> Component

The **<b:birt>** component servers to integrate a BIRT report into Seam or 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 4.1. <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.

4.2. <b:param> Component

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

Table 4.2. <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

You can find additional JBoss Developer Studio documentation at *RedHat documentation* [http://docs.redhat.com/docs/en-US/JBoss_Developer_Studio/index.html] website.

The latest documentation builds are available through the *JBoss Tools Nightly Docs Builds* [http://download.jboss.org/jbosstools/nightly-docs/].