

Space Details

Key:	Reporting
Name:	Pentaho Reporting Documentation - Latest
Description:	
Creator (Creation Date):	mdamour (Nov 27, 2006)
Last Modifier (Mod. Date):	admin (Nov 28, 2006)

Available Pages

- Integrating BIRT and Jasper
 - BIRT
 - 1. Eclipse BIRT Report Definitions
 - 2. Creating an Action Sequence With BIRT
 - 3. BIRT JDBC Driver Setup
 - 4. Verifying BIRT Integration in the Pentaho Platform
 - Jasper
 - 1. JasperReports Report Definition
 - 2. Creating an Action Sequence With Jasper
 - 3. Jasper JDBC Driver Setup
 - 4. Verifying JasperReports Integration in the Pentaho Platform

Integrating BIRT and Jasper

This page last changed on Dec 04, 2006 by [mdamour](#).

This guide demonstrates how to use the Pentaho BI Platform to run Birt and Jasper reports. The guide explains each type of report definition and steps you through creating an action sequence, setting up JDBC drivers, and verifying reports in the platform.

BIRT

This page last changed on Dec 04, 2006 by [mdamour](#).

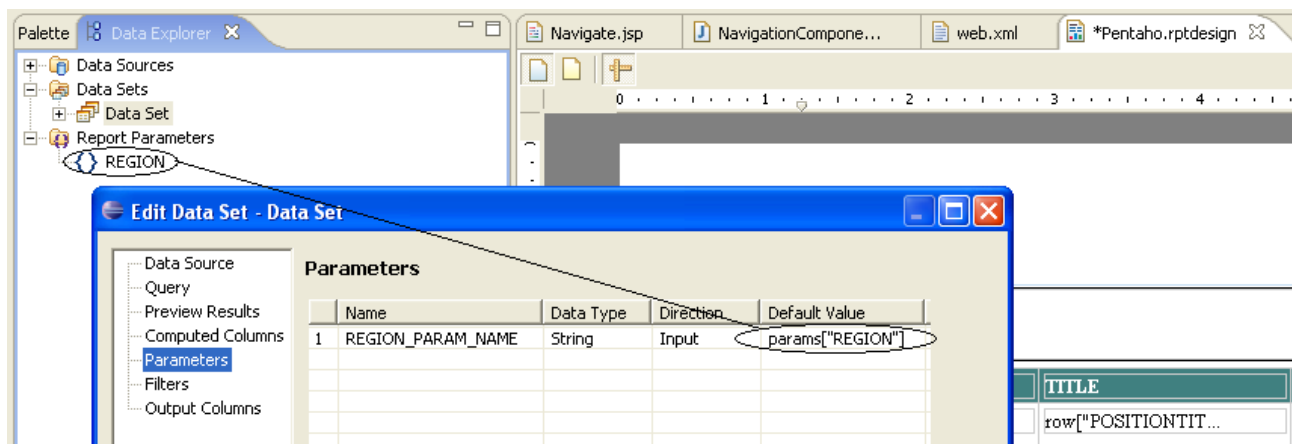
1. Eclipse BIRT Report Definitions

This page last changed on Dec 04, 2006 by [mdamour](#).

[2. Creating an Action Sequence With BIRT](#)

Eclipse BIRT report files (report.rptdesign) are simply XML documents with a funny extension. The extension allows the Eclipse IDE to recognize the file as a BIRT report. The Pentaho BIRTReportComponent (org.pentaho.plugin.eclipsebirt.BIRTReportComponent) is capable of executing these reports, and producing output as HTML, PDF, FO, and FOP.

Eclipse BIRT parameters are defined in the report definition as Scalar Parameters. In BIRT a parameter must be added to a Data Set. This is done when creating a Data Set or while editing an existing Data Set. There is a "Parameters" section of the Data Set editor dialog which is where you will define your parameters. When defining the parameter the "Default Value" must follow the convention `params["SOME_PARAMETER"]`. The "SOME_PARAMETER" must match the name of a report parameter as defined in BIRT's Data Explorer "Report Parameters" section.



For the example above, with a report parameter named "REGION", the report parameter has been defined as shown below.

Edit Parameter

Properties

Name:

REGION

Display name:

REGION

Type:

String

Display As

Format:

Unformatted

Change...

Display type:

Combo Box

Help text:

Selection values:

	Default	Value	Display Text
<input checked="" type="checkbox"/>	Default	Central	
<input type="checkbox"/>		Eastern	
<input type="checkbox"/>		Southern	
<input type="checkbox"/>		Western	
<input type="checkbox"/>		<Input new value here...>	
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

Import Values...

Set as Default

Delete

☒ Sort alphabetically when prompting

☐ Hidden

OK

Cancel

2. Creating an Action Sequence With BIRT

This page last changed on Dec 04, 2006 by [mdamour](#).

[1. Eclipse BIRT Report Definitions](#)

[3. BIRT JDBC Driver Setup](#)

After verifying that the report works in BIRT correctly, we can safely drop it into a Pentaho solution. Copy the BIRT report into {PCI}/pentaho-solutions/samples/reporting. What we need to do now is create a Pentaho action sequence XML document and save it in {PCI}/pentaho-solutions/samples/reporting. This document is made up of several sections: documentation, inputs, outputs, resources and action definitions. Note that the name of the action sequence document must match the <name> XML tag in it. Using the Pentaho Design Studio makes this easy. For complete details on how to use the Design Studio, download the documentation from Sourceforge at, http://sourceforge.net/project/showfiles.php?group_id=140317&package_id=171242.

Documentation; This section allows you to include author information, a description, an icon to represent the action sequence, a help URL and a result-type. For this example, we specify "report" as the result-type.

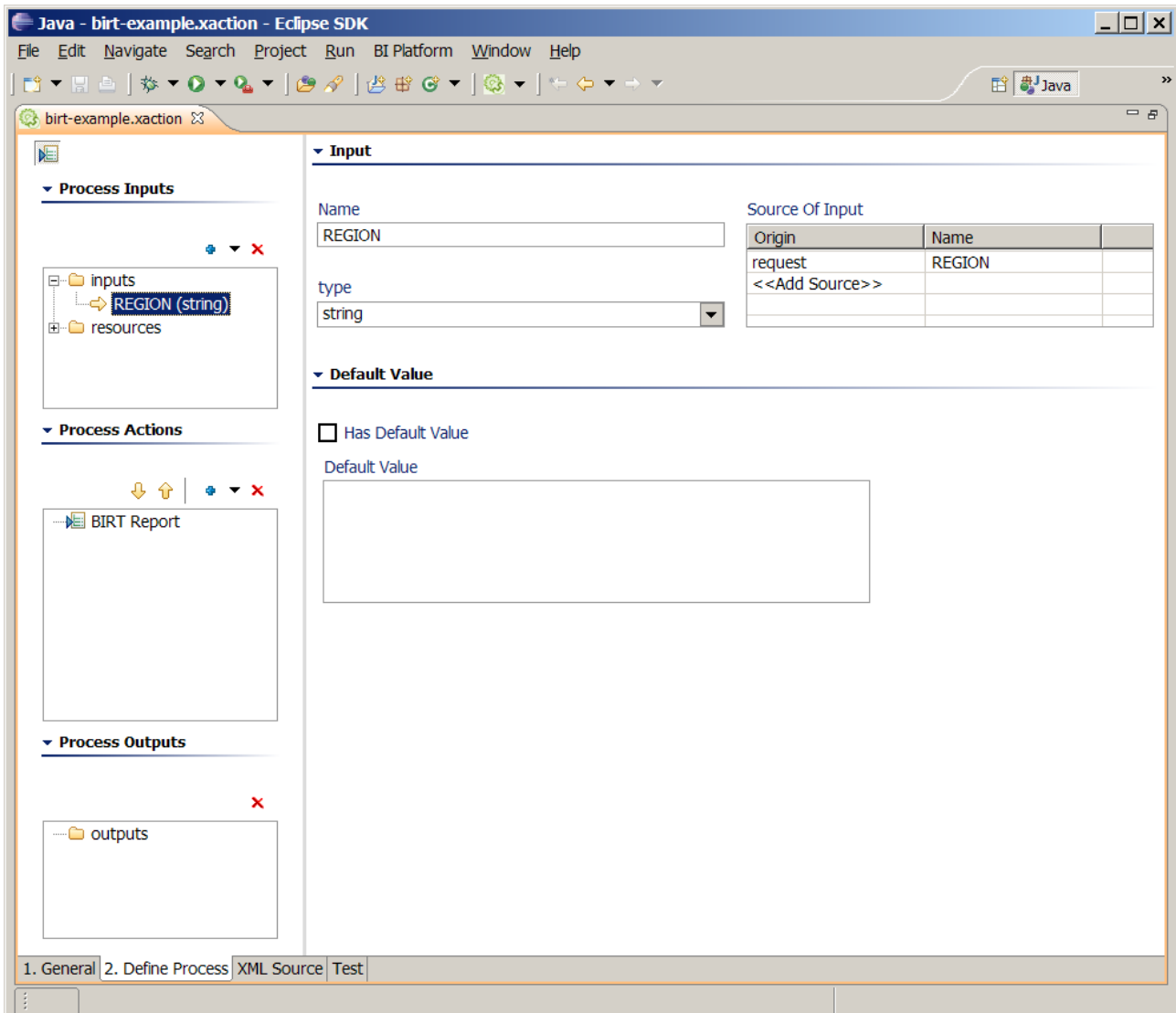
The screenshot shows the Eclipse SDK interface for editing a BIRT action sequence XML document. The window title is "Java - birt-example.xaction - Eclipse SDK". The menu bar includes File, Edit, Navigate, Search, Project, Run, BI Platform, Window, and Help. The toolbar contains various icons for file operations and development tools. The main editor area is titled "birt-example.xaction" and contains the following fields:

- Title:** Integrating Birt
- Version:** 1
- Logging Level:** ERROR (dropdown menu)
- Author:** Joe Pentaho
- Result Type:** (empty text field)
- Icon:** (empty box)
- Icon Location:** Browse... (button)
- Description:** Empty blank action sequence document (text area)
- Help:** (empty text area)

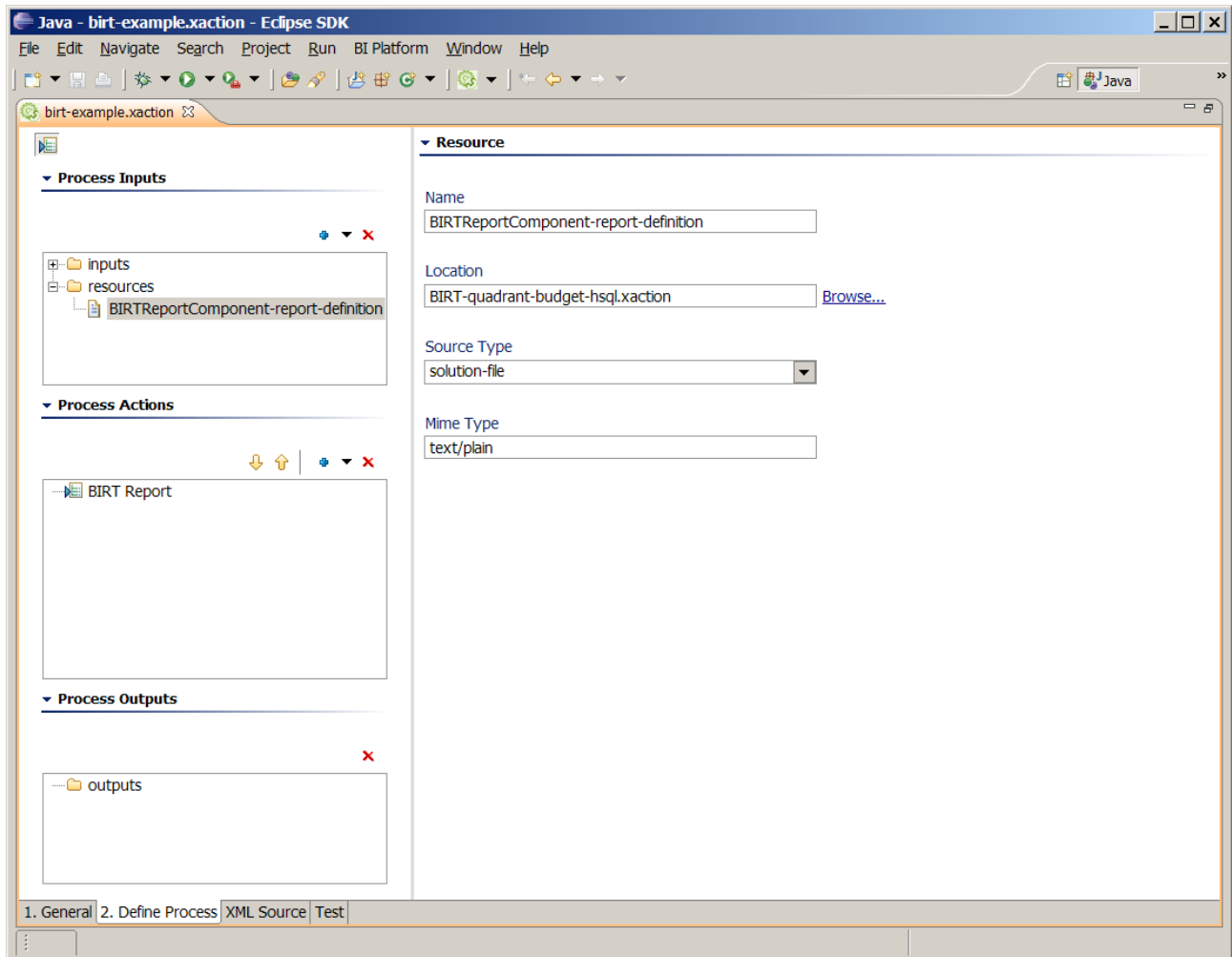
The bottom status bar shows four tabs: "1. General", "2. Define Process", "XML Source", and "Test". The "General" tab is currently selected.

Inputs; There are two inputs, output-type and REGION. Notice the REGION input is the same name as the report parameter from BIRT. This is important. The output-type in the example below is "html", other

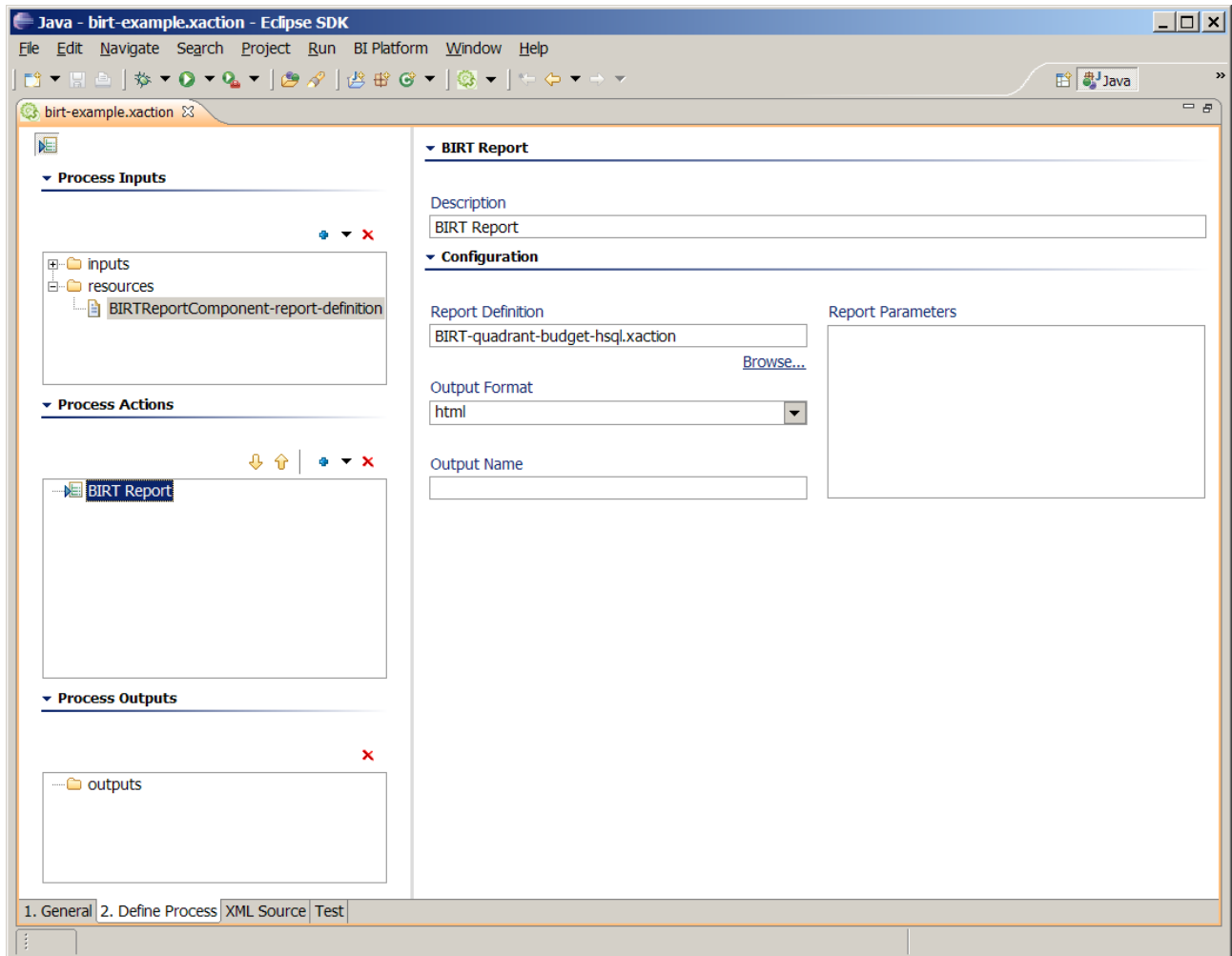
acceptable output-types are pdf, fo and fop.



Resources; In this section we define a "report-definition" which points to the BIRT .rptdesign XML file.



Actions; For this example we have only one action-definition. The component-name identifies the Java class that handles the action. For BIRT reports use name "BIRTReportComponent". There are two inputs to this action, the output-type and the REGION. These inputs are defined in the inputs section of the action-sequence document.



3. BIRT JDBC Driver Setup

This page last changed on Dec 04, 2006 by [mdamour](#).

[2. Creating an Action
Sequence With BIRT](#)

[4. Verifying BIRT
Integration in the Pentaho
Platform](#)

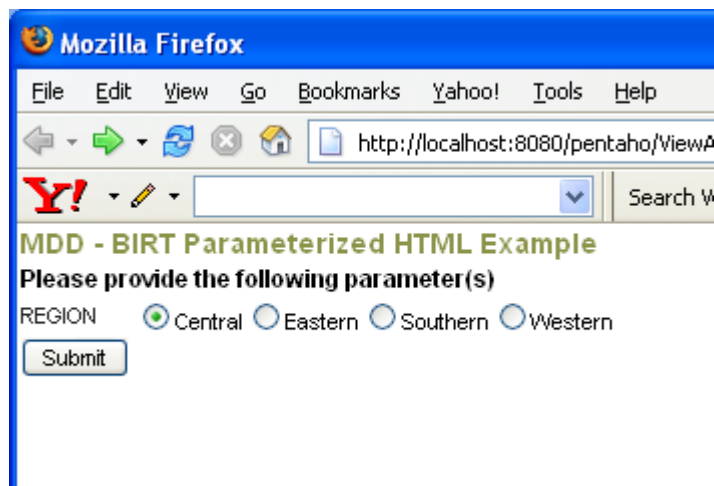
The BIRT report contains JDBC connection information that will be used by BIRT to generate the reports. Since we are generating the reports inside the Pentaho framework we must put the JDBC driver for the database we are using in {PCI}/jboss/server/default/lib.

4. Verifying BIRT Integration in the Pentaho Platform

This page last changed on Dec 04, 2006 by [mdamour](#).

[3. BIRT JDBC Driver Setup](#)

At this point the report should be plugged in and ready for use. Point your web browser to your PCI (typically <http://localhost:8080>). Navigate to the report that you created under the Reporting Examples group.



At this point you are prompted for the parameter.

A screenshot of a Mozilla Firefox browser window displaying a generated HTML report. The address bar shows a long URL: 'http://localhost:8080/pentaho/ViewAction?solution=samples&action=BIRT-MDD-region-hsql.action.xml&path=reporting%2FMDC'. The report content includes a heading 'Region: Central' and a table with five columns: DEPARTMENT, TITLE, ACTUAL, BUDGET, and VARIANCE. The table contains four rows of data for the Sales department, with variance values highlighted in red or yellow.

DEPARTMENT	TITLE	ACTUAL	BUDGET	VARIANCE
Sales	District Manager	\$682,625.00	\$617,250.00	-\$65,375.00
Sales	Senior Sales Rep	\$497,223.00	\$484,820.00	-\$12,403.00
Sales	Sales Rep	\$675,975.00	\$612,500.00	-\$63,475.00
Sales	Account Executive	\$409,975.00	\$422,500.00	\$12,525.00

Success! Here we see the generated report in HTML.

Jasper

This page last changed on Dec 04, 2006 by [mdamour](#).

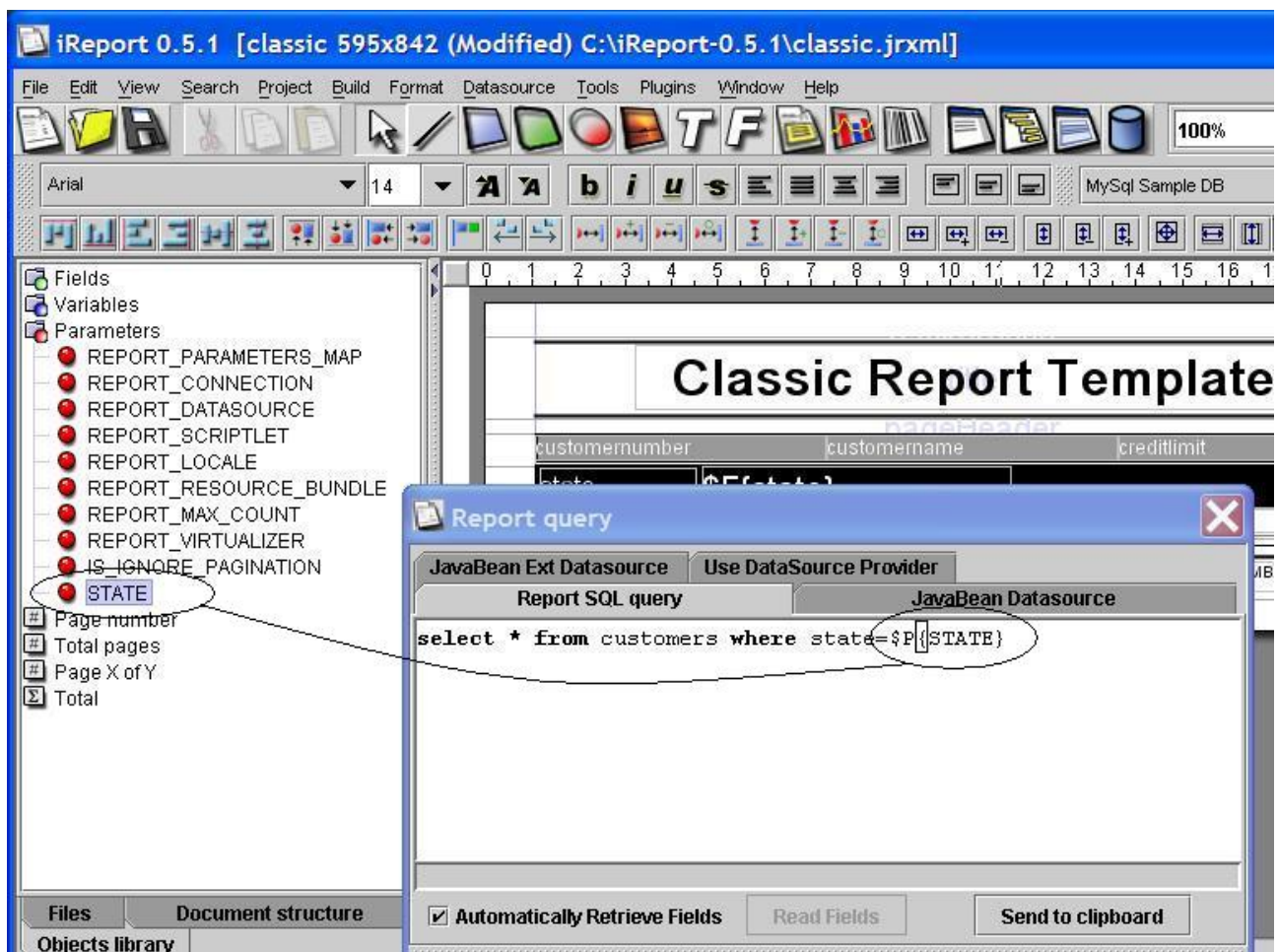
1. JasperReports Report Definition

This page last changed on Dec 04, 2006 by [mdamour](#).

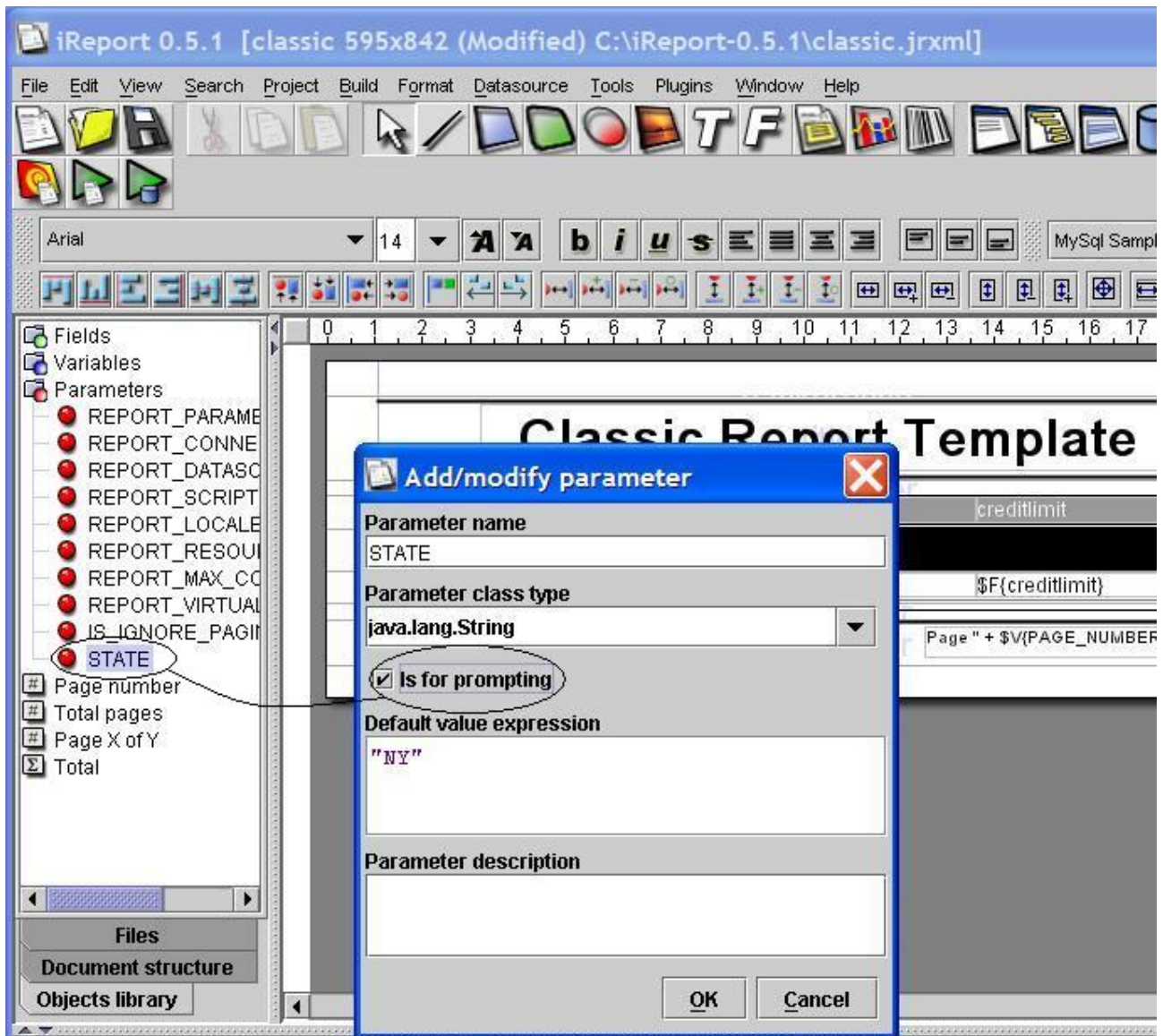
[2. Creating an Action Sequence With Jasper](#)

JasperReports report files (report.jrxml) are also just XML documents with a funny extension. In JasperReports though, the .jrxml file is considered the "source code" for your report. Before you can execute the report, you must compile the file into a file that has a .jasper extension. The Pentaho JasperReportsComponent (org.pentaho.plugin.jasperreports.JasperReportsComponent) is designed to do that "compilation" for you, so all you'll need to get started is the .jrxml file. The Pentaho JasperReportsComponent is able to execute these reports and produce output as either HTML or PDF.

JasperReports parameters are defined in the report definition. If you're using iReport to create your JasperReports, you create parameters in your Object Library then reference the parameter in your report query.



The report above is using a parameter named "STATE". In order for the user to be prompted for the value of the STATE parameter from within the Pentaho solution you must select "Is for prompting" when creating the parameter, as shown below.



2. Creating an Action Sequence With Jasper

This page last changed on Dec 04, 2006 by [mdamour](#).

[1. JasperReports Report Definition](#) [3. Jasper JDBC Driver Setup](#)

We can now drop the JasperReport into a Pentaho solution. Copy the jrxml file into {PCI}/pentaho-solutions/samples/reporting. What we need to do now is create a Pentaho action sequence XML document and save it in {PCI}/pentaho-solutions/samples/reporting. This document is made up of several sections: documentation, inputs, outputs, resources and action definitions. Note that the name of the action sequence document must match the <name> XML tag in it. Again, the Pentaho Design Studio makes this easy. For complete details on how to use the Design Studio, download the documentation from Sourceforge at, http://sourceforge.net/project/showfiles.php?group_id=140317&package_id=171242.

Documentation; This section allows you to include author information, a description, an icon to represent the action sequence, a help URL and a result-type. For this example, we specify "report" as the result-type.

The screenshot shows the 'Pentaho - untitled1.xaction - Eclipse SDK' window. The 'General' tab is active, displaying a form for configuring an action sequence. The form includes fields for Title, Version, Logging Level, Author, and Result Type, as well as an Icon field and an Icon Location field with a 'Browse...' button. The Description field contains the text 'Integrating Jasper example'. The Help field is empty. The bottom of the window shows a tabbed interface with '1. General', '2. Define Process', 'XML Source', and 'Test'.

Title	Icon	Icon Location
Integrating Jasper		Browse...

Version: 1

Logging Level: ERROR

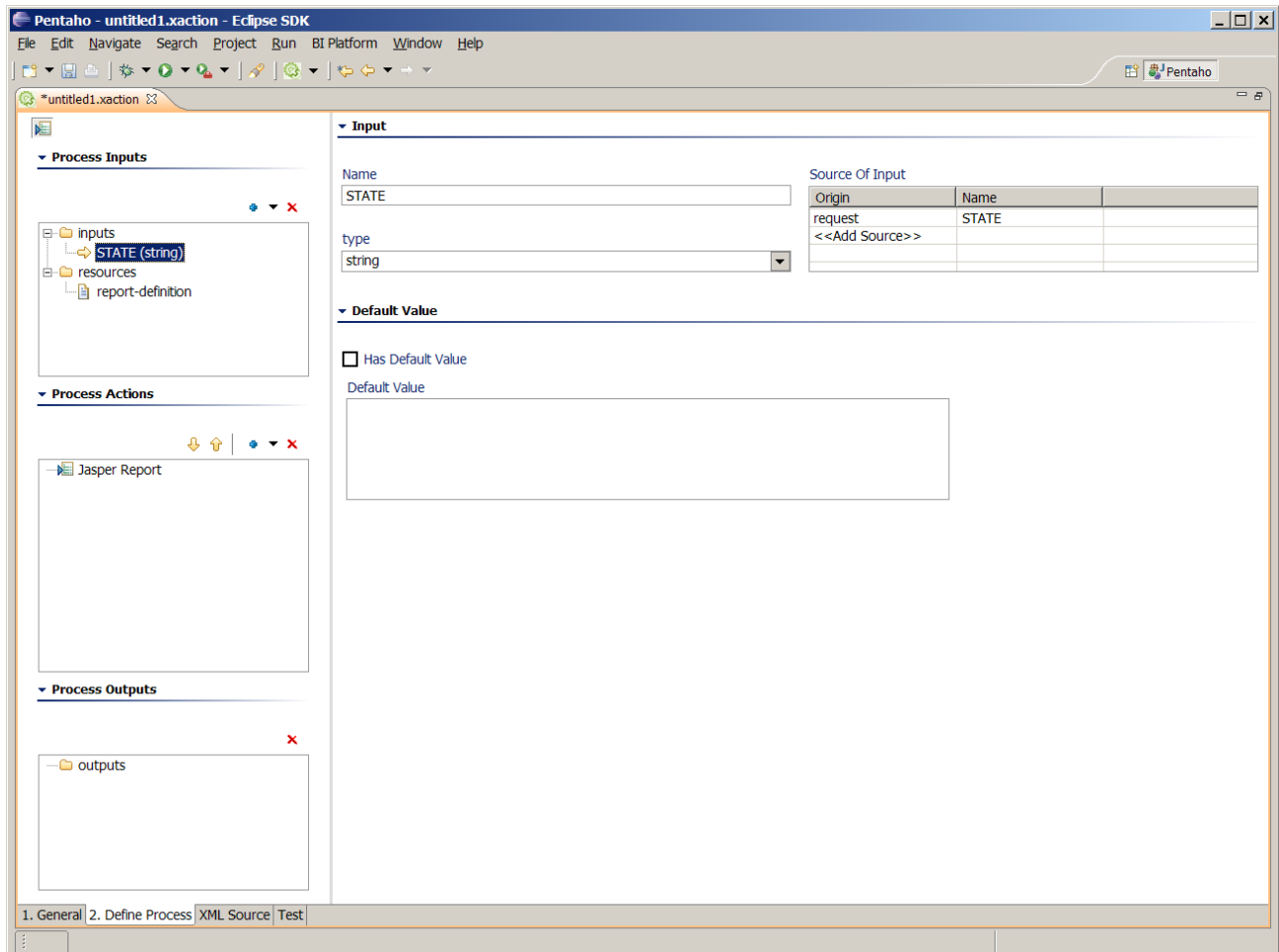
Author: Joe Pentaho

Result Type:

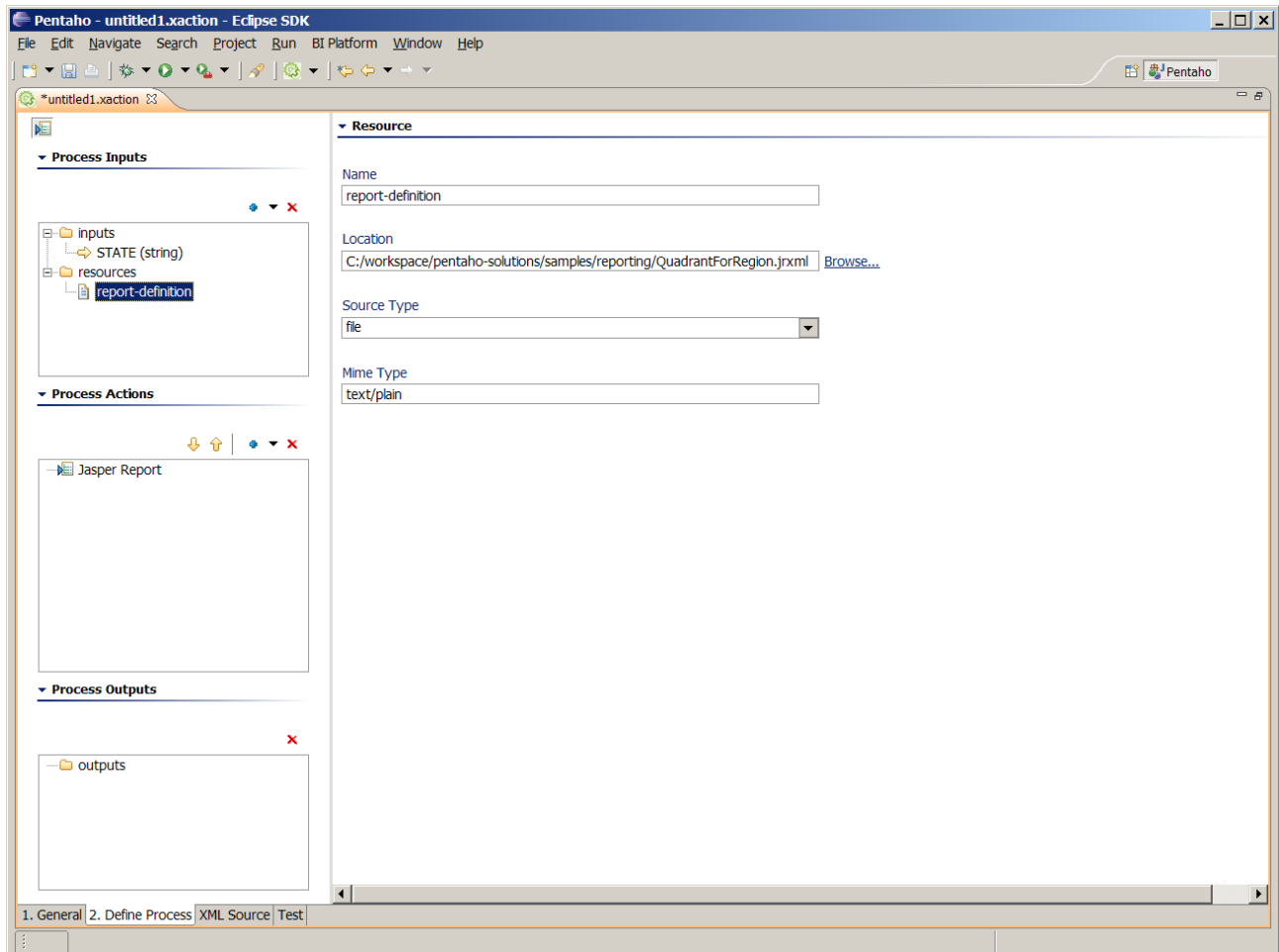
Description: Integrating Jasper example

Help:

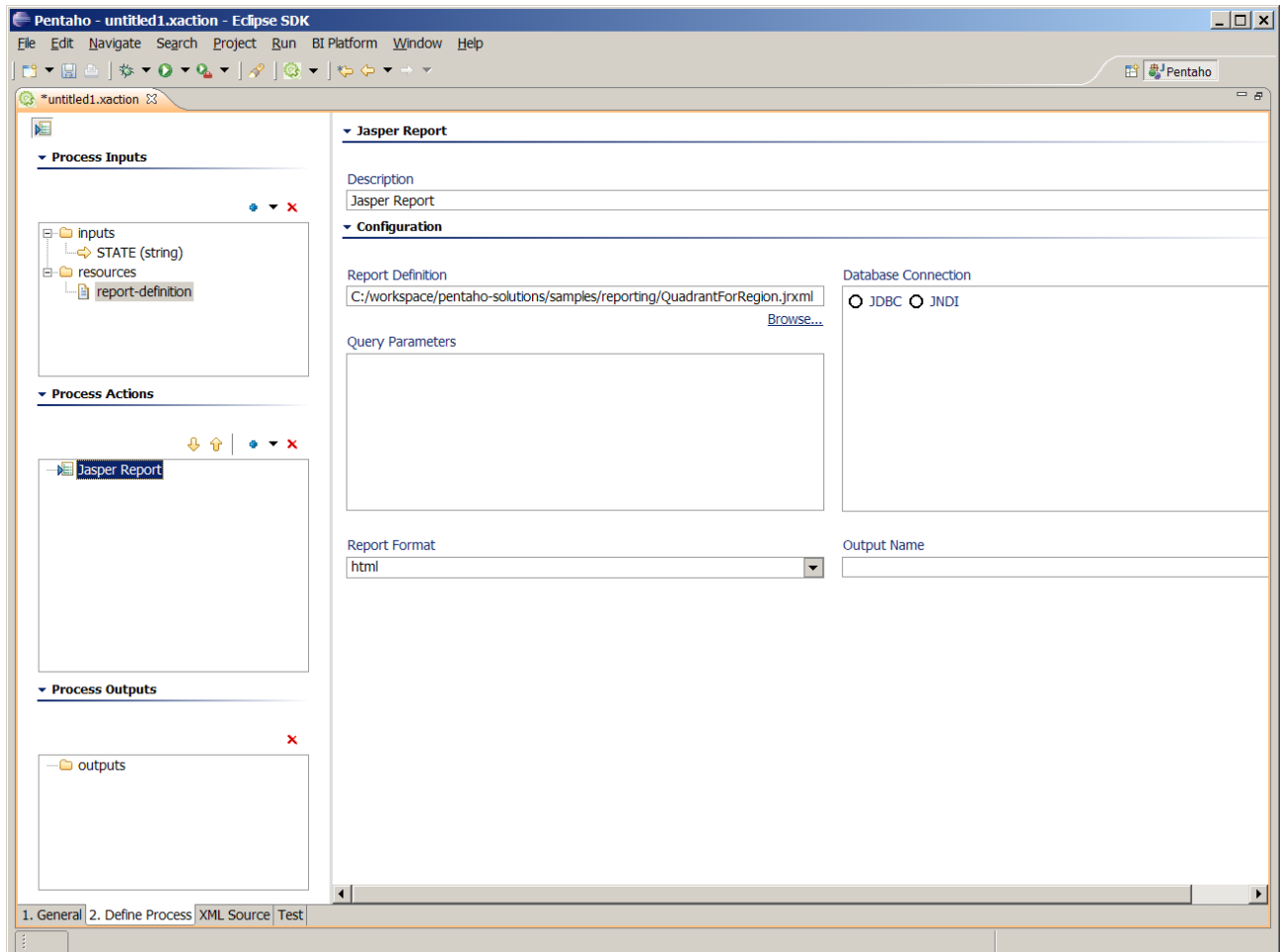
Inputs; There are two inputs, output-type and STATE. Notice the STATE input is the same name as the report parameter being used by the JasperReport. This is important. The output-type in the example below is "html". Optionally "pdf" could have been specified.



Resources; In this section we define a "report-definition" which points to the JasperReport .jrxml XML file.



Actions; For this example we have only one action-definition. The component-name identifies the Java class that handles the action. For JasperReports use "JasperReportsComponent". There are two inputs to this action, the output-type and the STATE. These inputs are defined in the inputs section of the action-sequence document.



3. Jasper JDBC Driver Setup

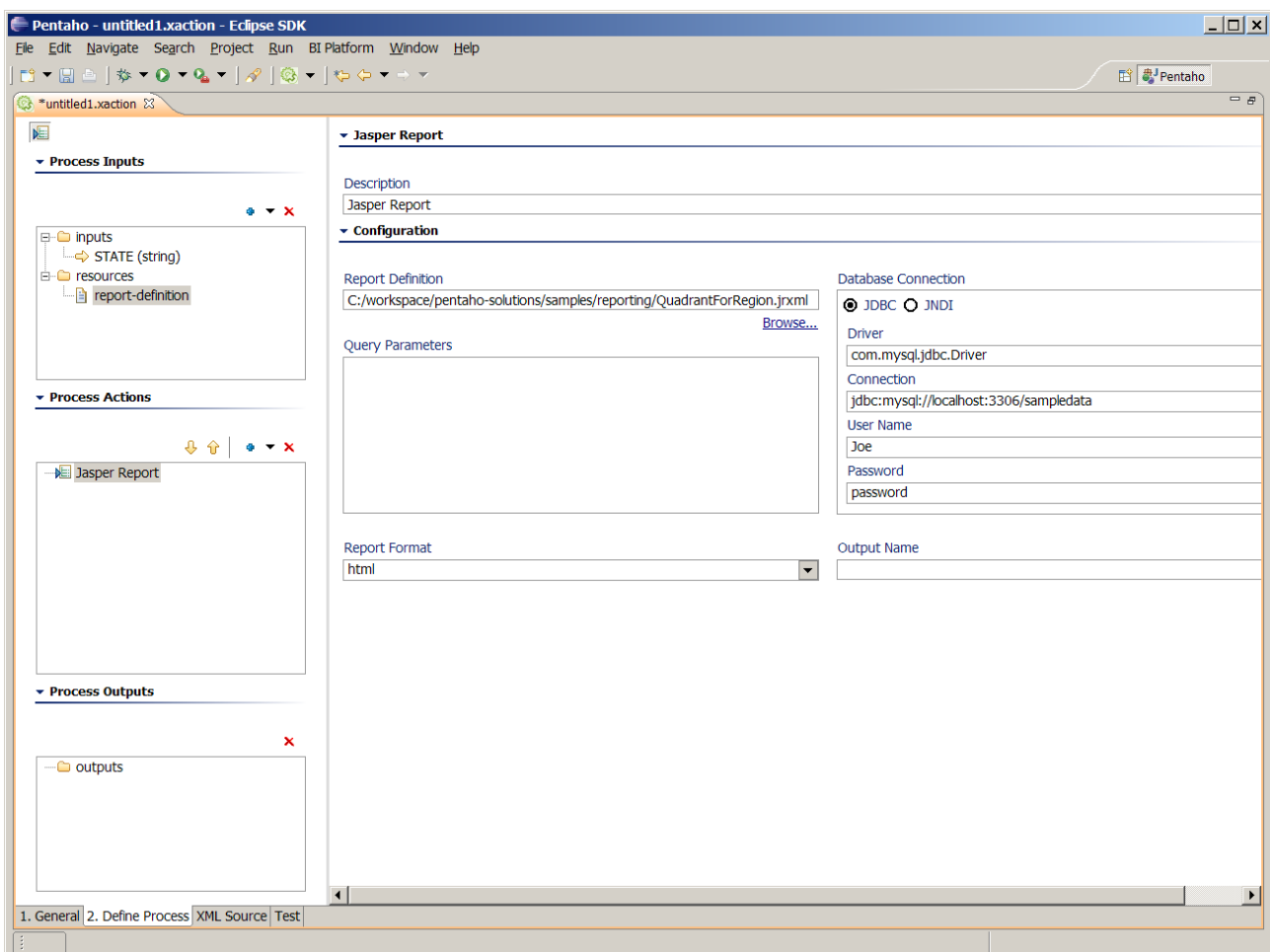
This page last changed on Dec 04, 2006 by [mdamour](#).

[2. Creating an Action Sequence With Jasper](#)

[4. Verifying JasperReports Integration in the Pentaho Platform](#)

Since we are generating the reports inside the Pentaho framework we must put the JDBC driver for the database we are using in {PCI}/jboss/server/default/lib.

Unlike BIRT report definitions, .jrxml files do not contain the database connection information for the report. This information needs to be specified in the action sequence. The database location can be defined within the action definitions as follows:



Alternatively the database can be identified using a JNDI name by simply clicking the JNDI radio button and entering a name. If you're going to use JNDI to identify the report database you'll need to configure JBoss to map the JNDI name to your database as follows:

- Create an xxxx-ds.xml file for your database type in {PCI}/jboss/server/default/deploy. For this example we'll create a mysql-ds.xml file with the following content.

```
<?xml version="1.0" encoding="UTF-8"?>

<datasources>
  <local-tx-datasource>
    <jndi-name>MySQLDS</jndi-name>
    <connection-url>jdbc:mysql://localhost:3306/sampledata</connection-url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <user-name>jim</user-name>
    <password>password</password>
    <exception-sorter-class-name>
      org.jboss.resource.adapter.jdbc.vendor.MySQLExceptionSorter
    </exception-sorter-class-name>
    <metadata>
      <type-mapping>mySQL</type-mapping>
    </metadata>
  </local-tx-datasource>
</datasources>
```

- Add a resource reference to {PCI}/jboss/server/default/deploy/pentaho.war/WEB-INF/web.xml.

```
<resource-ref>
  <description>MySQL Connection</description>
  <res-ref-name>jdbc/MySQLDS</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

- Add a resource reference
to {PCI}/jboss/server/default/deploy/pentaho.war/WEB-INF/jboss-web.xml.

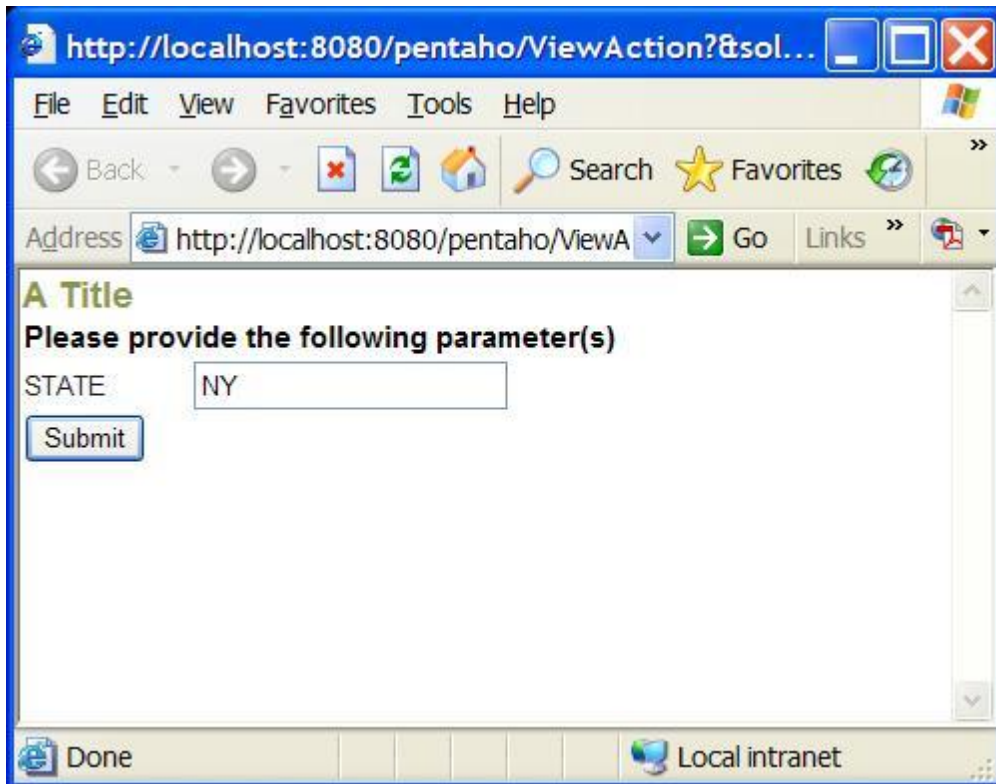
```
<resource-ref>
  <res-ref-name>jdbc/MySQLDS</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <jndi-name>java:/MySQLDS</jndi-name>
</resource-ref>
```

4. Verifying JasperReports Integration in the Pentaho Platform

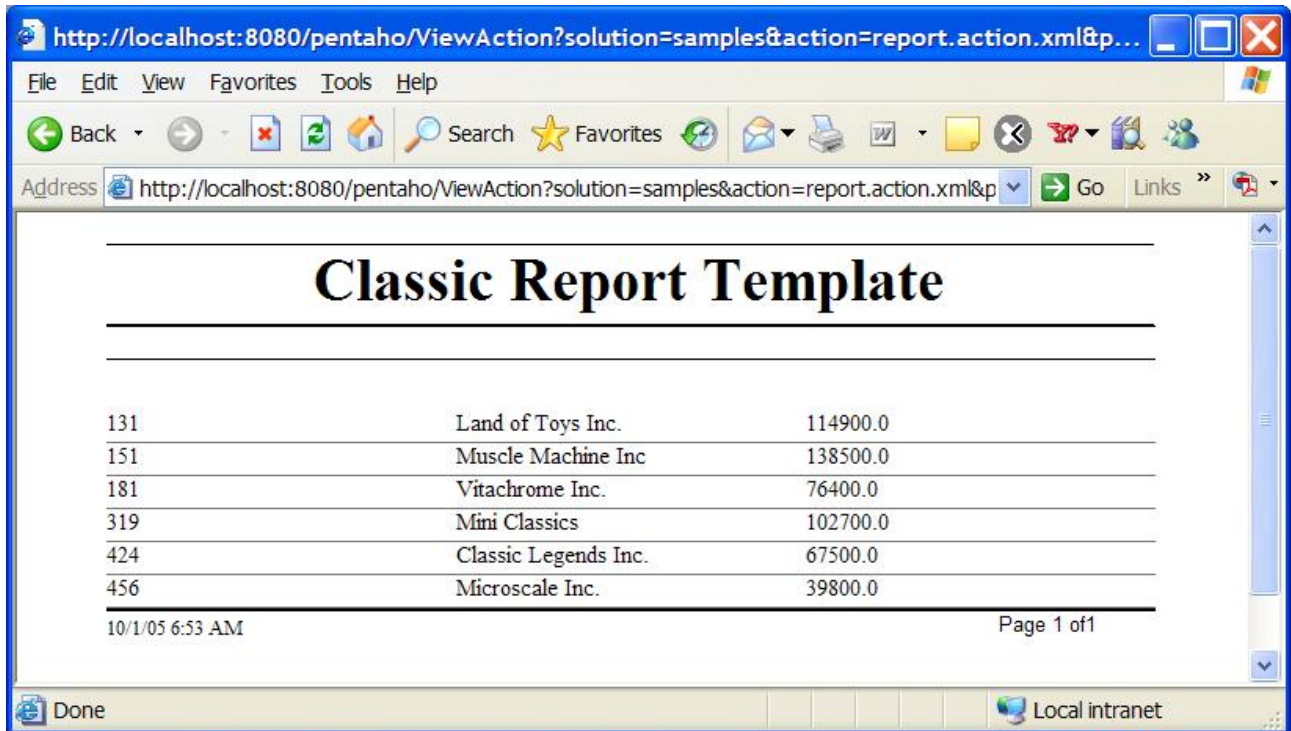
This page last changed on Dec 04, 2006 by [mdamour](#).

[3. Jasper JDBC Driver Setup](#)

Restart your Pentaho server by running {PCI}/stop_pentaho then {PCI}/start_pentaho. At this point the report should be plugged in and ready for use. Point your web browser to your PCI (typically <http://localhost:8080>). Navigate to the report that you created under the Reporting Examples group.



At this point you are prompted for the parameter.



At last we've reached our goal! Here we see the generated report in HTML. Isn't it a beauty?