

# Crystal Reports 10

## Getting Started with the Java Reporting Component (JRC)

---

### Overview

The Java Reporting Component (JRC) is bundled with Crystal Reports 10, Crystal Reports for Borland JBuilder X and Crystal Reports for BEA WebLogic Workshop 8.1. This document discusses how to perform a default installation of the JRC on all three Windows platforms, as well as provides code samples to display a report on these platforms.

### Contents

<b>INTRODUCTION</b> .....	<b>3</b>
<b>BEFORE YOU BEGIN</b> .....	<b>3</b>
<b>CRYSTAL REPORTS FOR BORLAND JBUILDER</b> .....	<b>3</b>
<i>Installing Crystal Reports for Borland JBuilder</i> .....	3
Installing from Borland JBuilder X.....	3
Installing from Crystal Reports 10.....	4
<i>Installing the Crystal Reports for Borland JBuilder Designer</i> .....	4
<i>Viewing a report with saved data using the viewer tag</i> .....	5
Create a project.....	5
Create a web module.....	5
Create a JSP file.....	5
<i>Viewing a report with saved data without using the viewer tag</i> .....	6
<b>CRYSTAL REPORTS FOR BEA WEBLOGIC WORKSHOP 8.1</b> .....	<b>8</b>
<i>Verifying the installation of the JRC</i> .....	8
<i>Installing the Crystal Reports for BEA WebLogic Designer</i> .....	8
<i>Viewing a report with saved data using the viewer tag</i> .....	9
<i>To view a report that contains saved data using the viewer tag in</i>	
<i>WebLogic:</i> .....	9
<i>Viewing a report with saved data without using the viewer tag</i> .....	10
<b>JAVA REPORTING COMPONENT FOR CRYSTAL REPORTS 10</b> .....	<b>11</b>
<i>Installing JRC from Crystal Reports 10</i> .....	11
<i>Viewing a report with saved data</i> .....	12
<i>Viewing a report that contains live data</i> .....	13
Using Java Naming and Directory Interface (JNDI).....	14
Using Java database connectivity (JDBC).....	14
<i>Passing parameter values to a report</i> .....	16
Main File.....	16
Viewer File.....	19
<i>Specifying report location in CrystalReportEngine-Config.xml</i> .....	19

**FINDING MORE INFORMATION ..... 21**  
*Configuring JNDI and JDBC data sources ..... 21*  
*JNDI name specified in the deployment descriptor file ..... 21*  
*Reporting off JDBC Data Sources ..... 21*

## Introduction

The Java Reporting Component (JRC) is a powerful Crystal reporting engine that takes full advantage of the benefits of Java, including portability across operating systems and hardware platforms. The JRC provides a fast, codeless way to connect to data, and to design dynamic, high fidelity presentation layers for J2EE-compliant applications.

## Before You Begin

Before proceeding through this document, ensure that you have setup and configured a Java application server such as Apache Tomcat. Since the Java Reporting Component (JRC) is written entirely in Java, only the Java application server is required to be running in order to view or export a report.

## Crystal Reports for Borland JBuilder

Crystal Reports for Borland JBuilder is bundled with Borland JBuilder X Enterprise Edition, and with Crystal Reports 10 Developer and Advanced Developer editions.

Crystal Reports for Borland JBuilder consists of the following components:

- Java Reporting Component (JRC)
- CrystalReportViewer
- Viewer Tag Library
- Integration JAR files

## Installing Crystal Reports for Borland JBuilder

### Installing from Borland JBuilder X

To install Crystal Reports for Borland JBuilder from Borland JBuilder X:

1. If you are performing a new and full installation of JBuilder, you will be prompted to insert the [Crystal Reports for Borland JBuilder CD](#). If you have already installed JBuilder without installing Crystal Reports for Borland JBuilder, then insert the [Crystal Reports for Borland JBuilder CD](#), and then double-click **Install.exe**.
2. In the list of languages, click the install language, click **OK**, and then click **Next**.
3. Click **I accept the terms of the License Agreement**, and then click **Next**.
4. Click **Full Installation**.

5. Click **Next** to install Crystal Reports for Borland JBuilder to the JBuilder home directory.
6. Click **Install**.
7. When the install completes, click **Done**.

Crystal Reports for Borland JBuilder is now installed.

## Installing from Crystal Reports 10

To install the JRC from Crystal Reports 10:

1. On the Crystal Reports 10 CD, double-click **CR10\_Autorun\_ENENT.exe**.
2. Click **Add Java Components**.
3. Click **Crystal Reports for Borland JBuilder**, and follow the instructions to complete the installation.

Crystal Reports for Borland JBuilder is now installed.

<b>NOTE</b>	By default, the Java Reporting Component JAR (Java archive) files are located in the following folders: <ul style="list-style-type: none"><li>▪ C:\Program Files\Common Files\Crystal Decisions\2.5\java\lib</li><li>▪ C:\Program Files\Common Files\Crystal Decisions\2.5\java\lib\external</li></ul>
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Installing the Crystal Reports for Borland JBuilder Designer

Now that you have installed Borland JBuilder X and Crystal Reports for Borland JBuilder, install the Crystal Reports for Borland JBuilder Designer as follows:

1. In the JBuilder IDE, edit a report that is in a project. A registration screen appears.
2. Register your copy of JBuilder. You are now redirected to a web page that contains a link to download the Crystal Reports for Borland JBuilder Designer.
3. Click the link, and follow the instructions to install the Crystal Reports Designer.

## Viewing a report with saved data using the viewer tag

To view a report using the **viewer** tag in Borland JBuilder X:

### Create a project

1. On the **File** menu, click **New Project**. The **Project Wizard** dialog box appears.
2. Type a name for the project in the **Name** box, and then click **Next**.
3. Click the **Required Libraries** tab, and click **Add**. The **Select One or More Libraries** dialog box appears.
4. Click **Crystal Reports Engine 10**, press and hold down the **Shift** key, and click **Crystal Reports Engine 10 Third Party**.
5. Click **OK** and then click **Finish** to create the project.

### Create a web module

1. On the **File** menu, click **New**. The **Object Gallery** dialog box appears.
2. Click **Web**, click **Web Module (WAR)**, and then click **OK**. The **Web Module Wizard** dialog box appears.
3. Click **Create empty Web Module** and then click **Next**.
4. Type the name for the module in the **Name** box, and then click **Next**.
5. Select the **Crystal Reports Engine 10 (Crystal Reports Viewer Tags 10)** check box, and then click **Finish**.

### Create a JSP file

1. On the **File** menu, click **New**. The **Object Gallery** dialog box appears.
2. Click **Web**, and then double-click **JavaServer Page**. The **JSP Wizard** dialog box appears.
3. Type the name of the JSP file in the **Name** box, and then click **Finish**.
4. In the JSP file, delete the tags that are between the **<body>** and **</body>** tags.
5. Place the insertion point between these two tags.
6. On the **Wizards** menu, click **Insert Crystal Reports Viewer**. The **Insert Crystal Reports Viewer Wizard** dialog box appears.

7. Type the name for the viewer in the **Name** box, and then click the **...** button. The **Specify a Crystal Report Location** dialog box appears.
8. Browse to a report file that contains saved data, click **OK**, and then click **Next**.
9. If this report was not previously added to the project, the **Report not in report directory** dialog box appears. In this case, click the directory that appears, and then click **OK**.
10. On the **Toolbar** tab, modify the viewer properties as desired, and then click **Finish**.
11. Save the JSP file, and then on the **Run** menu, click **Run Project**.

The report now appears in the viewer.

## Viewing a report with saved data without using the viewer tag

To view a report that contains saved data without using the **viewer** tag in Borland JBuilder X:

1. Complete the steps listed under [Create a project](#) and [Create a web module](#) in the previous section.
2. Complete steps 1 to 4 listed under [Create a JSP file](#) in the previous section.
3. On the **File** menu, click **New**. The **Object Gallery** dialog box appears.
4. Click **General**, and then double-click **Crystal Report**. The **New Crystal Reports Wizard** dialog box appears.
5. Change the name of the .rpt file, if desired.
6. Ensure that the **Launch Crystal Reports to edit the new report** checkbox is selected, and then click **OK**. The Crystal Reports Designer appears.
7. After creating the report, on the **File** menu, click **Save data with report**.
8. On the **File** menu, click **Save**.
9. Ensure that the report is added to the appropriate folder in your project.
10. Copy and paste the code below into the JSP file that was created in the section **Create a JSP File**:

**NOTE**

By default, the JRC searches for the report in the root folder of your web application (at the same hierarchical level as the WEB-INF folder). To specify a different folder, modify the file CrystalReportEngine-config.xml. For more information, refer to the section [Specifying Report Location in CrystalReportEngine-config.xml](#).

```

<%@ page language="java"
contentType="text/html;charset=UTF-8"%>

<%@ page
import="com.crystaldecisions.report.web.viewer.CrystalReportViewer,
com.crystaldecisions.reports.reportengineinterface.JPEReportSourceFactory,

com.crystaldecisions.sdk.occa.report.reportsource.IReportSourceFactory2,

com.crystaldecisions.sdk.occa.report.reportsource.IReportSource"%>

<html>
<head>
<title>
WithoutTag
</title>
</head>
<body bgcolor="#ffffff">
<%

/*create a path to the report and create a report source
off of it. The path must be a relative URL to the
report; it will not work to use the fully qualified URL or
pathname. Therefore the report must be in the same
directory as the web application or below.*/

/***** CHANGE FULL PATH TO YOUR REPORT *****/
String path = "<YOUR REPORT NAME.rpt>";
IReportSourceFactory2 rsf = new JPEReportSourceFactory();
IReportSource rptSource =
(IReportSource)rsf.createReportSource(path,
request.getLocale());
CrystalReportViewer viewer = new CrystalReportViewer();
viewer.setReportSource(rptSource);
viewer.processHttpRequest(request,
response,getServletConfig().getServletContext(), out);
%>
</body>
</html>

```

11. Save the JSP file, and then on the **Run** menu, click **Run Project**.

The report now appears in a viewer.

## Crystal Reports for BEA WebLogic Workshop 8.1

Crystal Reports for BEA WebLogic consists of the following components:

- Java Reporting Component (JRC)
- CrystalReportViewer
- Viewer Tag Library
- Integration JAR files

### Verifying the installation of the JRC

When you install BEA WebLogic Workshop 8.1, you also automatically install Crystal Reports for BEA WebLogic.

To verify that the components are correctly installed, in Windows Explorer:

1. Verify that the crystal-reports-10 folder exists in the following default location:

C:\bea\weblogic81\workshop\thirdparty\crystal-reports-10

2. Verify that the crystalreportviewers10 and java\lib folders exist in the following default location:

C:\bea\weblogic81\common\eval\crystalreports2.5\

### Installing the Crystal Reports for BEA WebLogic Designer

Now that you have verified the installation of Crystal Reports for BEA WebLogic, install the Crystal Reports for BEA WebLogic Designer as follows:

1. Complete steps 1 to 10 in the section [Viewing a report with saved data using the viewer tag](#) to create a project that contains a report.
2. In the BEA WebLogic Workshop IDE, double-click the report that is in your project. A registration screen appears.
3. Complete the registration form. You are now redirected to a link to download Crystal Reports for BEA WebLogic Workshop.
4. Click the link, and follow the instructions to install the Crystal Reports Designer.

## Viewing a report with saved data using the viewer tag

To view a report that contains saved data using the **viewer** tag in WebLogic:

1. On the **File** menu, click **New**, and then click **Application**. The **New Application** dialog box appears.
2. Click **Default Application** in the list of project types.
3. In the **Name** box, type a name for the application, and then click **Create**. You have now created a web application folder named <your application name>Web.

For example: An application called Test creates a web application called TestWeb. TestWeb is visible as a folder in the context of the main application folder, and contains a WEB-INF and resources folder, as well as a Controller.jspf, error.jsp, and index.jsp file.

4. On the **Application** tab, right-click the web application folder, click **Install**, and then click **Crystal Reports**. The JRC JAR files, the Viewer Tag Library, and other Crystal-specific resources are now added to the web application.
5. Expand the web application folder, right-click **index.jsp**, click **Open**, and then delete the text “New Web Application Page”.
6. On the **Palette** tab, in the section **Crystal Decisions**, drag and drop the **Viewer** icon onto the **index.jsp** workspace. The **Insert Crystal Reports Viewer Wizard** appears.

<b>NOTE</b>	If the <b>Palette</b> tab is not visible, on the <b>View</b> menu, click <b>Windows</b> , and then click <b>Palette</b> .
-------------	---------------------------------------------------------------------------------------------------------------------------

7. Click the **...** button. The **Open** dialog box appears.
8. Browse to and click a report (with saved data), click **Open**, and then Click **Next**.
9. If this report is not already added to the project, the **Report not in report directory** dialog box appears. Click **OK**.
10. Set the desired properties for the viewer, and then click **Finish**.
11. On the **Debug** menu, click **Start**.
12. You may receive a message saying your web page is part of a page flow and Controller.jspf will be run. Click **OK** to continue.

The report now appears in the viewer.

## Viewing a report with saved data without using the viewer tag

To view a report that contains saved data without using the **viewer** tag in WebLogic:

1. Complete steps 1 to 4 listed under [Viewing a report with saved data using the viewer tag](#).
2. On **File** menu, click **Import Files**. The **Import Files to Project '<Name of your Project>'** appears.
3. Browse to and click a report that contains saved data, and then click **Import**. The report is now added to the application.

### NOTE

By default, the JRC searches for the report in the root folder of your web application (at the same hierarchical level as the WEB-INF folder) . To specify a different folder, modify the file CrystalReportEngine-config.xml. For more information, refer to the section [Specifying Report Location in CrystalReportEngine-config.xml](#).

4. Expand the name of the application folder, right-click **index.jsp**, click **Open**.
5. Switch to the source view of index.jsp by clicking on **Source View** at the bottom of the workspace window.
6. Delete the code in the index.jsp file and add the following code, setting the **path** variable to the appropriate value:

```
<%@ page language="java"
contentType="text/html;charset=UTF-8"%>

<%@ page
import="com.crystaldecisions.report.web.viewer.CrystalRe
portViewer"%>

<%@ page
import="com.crystaldecisions.reports.reportengineinterfa
ce.JPEReportSourceFactory"%>

<%@ page
import="com.crystaldecisions.sdk.occa.report.reportsourc
e.IReportSourceFactory2"%>

<%@ page
import="com.crystaldecisions.sdk.occa.report.reportsourc
e.IReportSource" %>

<head>
<title>Web Application Page</title>
</head>
<body>
<%
```

```
/*create a path to the report and create a report source
off of it. The path must be a relative URL to the
```

report; it will not work to use the fully qualified URL or pathname. Therefore the report must be in the same directory as the web application or below.\*/

```
String path = "<YOUR REPORT NAME.rpt>";

IReportSourceFactory2 rsf = new
JPEReportSourceFactory();
IReportSource rptSource =
(IReportSource)rsf.createReportSource(path,
Locale.ENGLISH);

CrystalReportViewer viewer = new CrystalReportViewer();
viewer.setReportSource(rptSource);
viewer.processHttpRequest(request, response,
getServletConfig().getServletContext(), out);

%>

</body>
```

7. On the **Debug** menu, click **Start**.
8. You may receive a message saying your web page is part of a page flow and Controller.jspf will be run. Click **OK** to continue.

The report now appears in the viewer.

## Java Reporting Component for Crystal Reports 10

The Java Reporting Component (JRC) is also included with the Developer and Advanced Developer Editions of Crystal Reports 10. These versions allow the development of JRC applications, much like what can be accomplished with Crystal Reports for Borland JBuilder X and Crystal Reports BEA Weblogic Workshop 8.1.

### Installing JRC from Crystal Reports 10

To install the Java Reporting Component (JRC) that is included with Crystal Reports 10:

1. First, download and install the latest [Java JDK](#) and an application server such as [Apache Tomcat](#).
2. On the Crystal Reports CD, double-click **CR10\_Autorun\_ENENT.exe**.
3. Click **Add Java Components**.
4. Click **Crystal Reports 10 Java Reporting Component**, and follow the installation instructions. When prompted to enter a key code, use the same key code that you used to install Crystal Reports 10.

The JRC is now installed. The JRC JAR (Java archive) files are located in the following default folder:

C:\Program Files\Common Files\Crystal Decisions\2.5\java\lib

## Viewing a report with saved data

Now that the Java Reporting Component (JRC) is installed, to view a report that has been saved with data:

1. Copy the JAR files in the Program Files\Common Files\Crystal Decisions\2.5\java\lib folder to the WEB-INF\lib folder in your web application.
2. Copy the JAR files in the Program Files\Common Files\Crystal Decisions\2.5\java\lib\external folder to the WEB-INF\lib folder in your web application.
3. Create a folder called classes in the WEB-INF folder in your web application folder. If you did not create a custom web application folder, then the classes folder should be created under the WEB-INF folder in the root context of the application server.
4. Copy the CrystalReportEngine-config.xml and log4j.properties files into the classes folder. These files are located in \Program Files\Common Files\Crystal Decisions\2.5\java\lib folder.
5. Copy the crystalreportviewers10 folder from Program Files\common files\Crystal Decisions\2.5 to the root folder of your web application. If you did not create a custom web application folder, then save the crystalreportviewers10 folder in the root context of the application server.
6. Create a report, and save it with data.
7. Ensure that the report is added to the appropriate folder in your project.

<b>NOTE</b>	<p>By default, the JRC searches for the report in the root folder of your web application (at the same hierarchical level as the WEB-INF folder) . To specify a different folder, modify the file CrystalReportEngine-config.xml. For more information, refer to the section <a href="#">Specifying Report Location in CrystalReportEngine-config.xml</a>.</p>
-------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

8. Create a new JSP file, and add the following code to it, setting the path variable to the relative URL to the appropriate report:

```
<%@ page language="java"
contentType="text/html;charset=UTF-8"%>
<% page
import="com.crystaldecisions.report.web.viewer.CrystalReportViewer"%>
<%@ page
```

```

import="com.crystaldecisions.reports.reportengineinterface.
JPEReportSourceFactory,

com.crystaldecisions.sdk.occa.report.reportsource.IReportSo
urceFactory2,

com.crystaldecisions.sdk.occa.report.reportsource.IReportSo
urce"%>

<html>
<head>
<title>
WithoutTag
</title> </head>
<body bgcolor="#ffffff">

<%
/*create a path to the report and create a report source
off of it. The path must be a relative URL to the
report; it will not work to use the fully qualified URL or
pathname. Therefore the report must be in the same
directory as the web application or below.*/

String path = "<YOUR REPORT NAME.rpt>";
IReportSourceFactory2 rsf = new
JPEReportSourceFactory();
IReportSource rptSource =

(IReportSource)rsf.createReportSource(path,request.getLocal
e());

CrystalReportViewer viewer = new CrystalReportViewer();
viewer.setReportSource(rptSource);
viewer.processHttpRequest(request,
response,getServletConfig().getServletContext(), out);
%>
</body>
</html>

```

9. Ensure that the JSP file is saved under the root folder of your web application. If you did not create a custom web application folder, then the JSP file should be saved under the root context of your application server.

10. Start the application server and launch the JSP file in a web browser.

The report now appears in the viewer.

## Viewing a report that contains live data

When reports that do not have data saved with them are previewed, they access the data source to retrieve the data. This section discusses how to connect reports to their data sources, as well as how to pass parameter values. While the

connection methods are specific to Microsoft Access and Microsoft SQL Server 2000, the connection methodologies may be applied when connecting to other databases.

Since the Java Reporting Component (JRC) is a pure Java reporting engine, it can only access data sources using JNDI (Java Naming and Directory Interface) or JDBC (Java Database Connectivity) data sources.

<b>NOTE</b>	<p>While JDBC and JNDI data source connectivity is installed with the Crystal Reports BEA WebLogic Designer and the Crystal Reports for JBuilder Designer, it is not included with the Crystal Reports 10 Designer.</p> <p>To include JDBC and JNDI data source connectivity in Crystal Reports 10:</p> <ol style="list-style-type: none"><li>1. Click this link: <a href="http://www.businessobjects.com/products/downloadcenter/crystalreports.asp">http://www.businessobjects.com/products/downloadcenter/crystalreports.asp</a></li><li>2. Click <b>Download Windows JDBC and DB2 Unicode Drivers – all languages</b>.</li><li>3. Complete the remaining installation steps.</li></ol>
-------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Using Java Naming and Directory Interface (JNDI)

It is possible with the Java Reporting Component (JRC) to connect through JDBC/JNDI to a report that accesses an ODBC or OLE DB data source, provided that you have created a JNDI data source with the same name as the ODBC or OLE DB data source.

When the JRC application previews an ODBC or OLEDB report, it searches for a JNDI data source with the same name as the ODBC or OLE DB data source. For example, a report is created that uses an ODBC DSN called *Xtreme Sample Database 10*. When the JRC application previews the report, it searches for a JNDI resource also called *Xtreme Sample Database 10*.

While it is recommended that the JNDI name is configured on the Web server, in the case of an ODBC report, you also have the option of only specifying it in the web.xml (deployment descriptor) file. See [For More Information](#) for additional resources on setting up a JNDI resource on a web server and on specifying the JNDI resource in the deployment descriptor file.

## Using Java database connectivity (JDBC)

If you wish to view a report that contains live data, it is not necessary to configure a JNDI connection pool on your Web server or to specify a connection string in the web.xml file. With a JDBC connection, you are either prompted for the user name and password or you can set this information in your application.

The following steps demonstrate how to view a report that uses a JDBC connection to a Microsoft SQL Server database:

1. Go to the Microsoft web site at <http://www.microsoft.com>
2. Search for the following:

SQL Server 2000 Driver for JDBC

3. Download and install the SQL Server 2000 Driver for JDBC.
4. Copy the following JAR files from the location you installed the SQL Server 2000 Driver for JDBC to the WEB-INF/lib folder of your application:
  - msbase.jar
  - msutil.jar
  - mssqlserver.jar
5. In the Crystal Reports Designer, create a report that uses a JDBC connection, verifying that you have the correct JDBC driver name and URL to connect to the data source. For more information, refer to the section, [Finding More Information](#) at the end of this document.
6. If you are connecting to a secured database, use this sample code to provide the logon credentials to your database:

```
<%@ page
language="java"contentType="text/html;charset=UTF-8"%>

<%@ page import=
"com.crystaldecisions.report.web.viewer.*,com.crystaldec
isions.sdk.occa.report.data.*,
com.crystaldecisions.reports.reportengineinterface.JPERe
portSourceFactory,
com.crystaldecisions.sdk.occa.report.reportsource.IRepor
tSourceFactory2,
com.crystaldecisions.sdk.occa.report.reportsource.IRepor
tSource"%>

<html>
<head>
<title>
DatabaseLogon
</title>
</head>
<body bgcolor="#ffffff">
<%

/*create a path to the report and create a report source
off of it. The path must be a relative URL to the
report; it will not work to use the fully qualified URL
or pathname. Therefore the report must be in the same
directory as the web application or below.*/

String path = "<YOUR REPORT NAME.rpt>";
```

```

        IReportSourceFactory2 rsf = new
        JPEReportSourceFactory();

        IReportSource rptSource =
        (IReportSource)rsf.createReportSource(path,
        request.getLocale());

        ConnectionInfos connInfos = new ConnectionInfos();
        IConnectionInfo connInfo1 = new ConnectionInfo();
        connInfo1.setUserName("<YOUR USER NAME>");
        connInfo1.setPassword("<YOUR PASSWORD>");
        connInfos.add(connInfo1);

        CrystalReportViewer viewer = new
        CrystalReportViewer();

        viewer.setDatabaseLogonInfos(connInfos);
        viewer.setReportSource(rptSource);
        viewer.setEnableLogonPrompt(false);

        viewer.processHttpRequest(request, response,
        getServletConfig().getServletContext(),
        out);
    %>
</body>
</html>

```

## Passing parameter values to a report

The following Java Server Pages (JSP) code demonstrates how to pass string and number parameter values to a report. The code consists of two JSP files, a main file and a viewer file

### Main File

```

<%@ page language="java"
contentType="text/html;charset=UTF-8"%>
<%@ page import= "com.crystaldecisions.report.web.viewer.*,
com.crystaldecisions.sdk.occa.report.data.*"%>
<%@ page import="
com.crystaldecisions.reports.reportengineinterface.JPERepor
tSourceFactory,
com.crystaldecisions.sdk.occa.report.reportsource.IReportSo
urceFactory2,
com.crystaldecisions.sdk.occa.report.reportsource.IReportSo
urce"%>
<html>

```

```
<head>
    <title>
        Web Application Page
    </title>
</head>
<body>
<%
/*create a path to the report and create a report source
off of it. The path must be a relative URL to the
report; it will not work to use the fully qualified URL or
pathname. Therefore the report must be in the same
directory as the web application or below.*/
String path = "<YOUR REPORT NAME.rpt>";
IReportSourceFactory2 rsf = new JPEReportSourceFactory();
IReportSource rptSource =
(IReportSource)rsf.createReportSource(path,
request.getLocale());

//Create parameterfield and value objects to add to the
report's field collection
Fields fields = new Fields();
ParameterField pfield1 = new ParameterField();
ParameterField pfield2 = new ParameterField();
Values vals1 = new Values();
Values vals2 = new Values();
ParameterFieldDiscreteValue pfieldDV1 = new
ParameterFieldDiscreteValue();
        ParameterFieldDiscreteValue pfieldDV2 = new
ParameterFieldDiscreteValue();

/*Set the name of the report to pass the paramters to; this
is used to distinguish between the main report and
subreports. If passing to the main report, the name is
blank; if passing to the subreport, use the subreport
name*/
pfield1.setReportName("");

//This is the name of the parameter in the
reportpfield1.setName("stringParameter");

//The value you set for your parameter
pfieldDV1.setValue("This is a test");

//This is the parameter prompt
```

```
pfieldDV1.setDescription("Passing string parameter to the
report");
vals1.add(pfieldDV1);
pfield1.setCurrentValues(vals1);
Integer number = new Integer("5");
pfield2.setReportName("");

//This is the name of the parameter in the report
pfield2.setName("numberParameter");
pfieldDV2.setValue(number);

//This is the parameter prompt
pfieldDV2.setDescription("Passing number parameter to the
report");
vals2.add(pfieldDV2);
pfield2.setCurrentValues(vals2);

//add the parameter fields to the report fields collection
fields.add(pfield1);
fields.add(pfield2);

//Puts the objects in a session variable to be passed to
the viewer page
session.setAttribute("fields", fields);
session.setAttribute("rptSource", rptSource);
//Redirects users to the viewer page
response.sendRedirect("viewer.jsp");

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

## Viewer File

```

<%@ page language="java"
contentType="text/html;charset=UTF-8"%>
<%@ page import= "com.crystaldecisions.report.web.viewer.*,
com.crystaldecisions.sdk.occa.report.data.*,
com.crystaldecisions.reports.reportengineinterface.JPERepor
tSourceFactory,
com.crystaldecisions.sdk.occa.report.reportsource.IReportSo
urceFactory2,
com.crystaldecisions.sdk.occa.report.reportsource.IReportSo
urce"%>

<%
//Get the objects from session
IReportSource rptSource = (IReportSource)
session.getAttribute("rptSource");
Fields fields = (Fields) session.getAttribute("fields");

CrystalReportViewer viewer = new CrystalReportViewer();
viewer.setReportSource(rptSource);
viewer.setParameterFields(fields);
viewer.processHttpRequest(request,
response,getServletConfig().getServletContext(), out);
%>

```

## Specifying report location in CrystalReportEngine-Config.xml

You must specify the report location by setting the value of the **report location** tag in the CrystalReportEngine-config.xml file, which is located in the WEB-INF/classes folder. This tag specifies a path to the report that is relative to the location of the Java Reporting Components (JRC) JAR files. The default location of these JAR files is in the WEB-INF/lib folder of your web application. By default, the **report location** tag in the CrystalReportEngine-config.xml file is set to the following value:

```
../..
```

This default setting specifies that reports are stored in the root context of your web application, or in one of its subfolders. For example, if the report is saved under the root context, the JSP file should specify the report path as

```
String path = "/<ReportName>";
```

For reports saved in a subfolder of the root context, the JSP file specifies the report path as

```
String path = "/<SubFolder name>/<ReportName>";
```

**NOTE**

To highlight text in a PDF document for copying and pasting code, click the **Text Select Tool** toolbar button in Adobe Acrobat.



This procedure applies to Adobe Acrobat 4.0 and 5.0.

**DISCLAIMER**

The products discussed in this white paper have been fully tested in conjunction with Java Reporting Component prior to its release. Officially supported JRC platforms are listed in the text file Platforms.txt found in the root folder of the Crystal Reports CD.

The Information in this document is provided as a courtesy to assist our customers with the configuration of our product with supported third party platforms.

In the event issues arise with an unsupported configuration, there is no escalation support; however, they will be considered during the development of the next generation of our product.

## Finding More Information

### Configuring JNDI and JDBC data sources

For more information about configuring JNDI and JDBC data sources, search for cr10\_jrc\_data\_sources.pdf on our [support site](#).

### Specifying JNDI name in the deployment descriptor file

For more information on how to specify the JNDI name in the deployment descriptor file (web.xml), search for cr10\_jrc\_data\_sources.pdf on our [support site](#).

### Reporting off JDBC Data Sources

For more information on how to report off of a JDBC data source from Crystal Reports, search for “JDBC Data Sources” in the Help file that accompanies your product.

For more information and resources, refer to the product documentation and visit the support area of the web site at:

▶ [support.businessobjects.com](http://support.businessobjects.com)

The Business Objects product and technology are protected by US patent numbers 5,555,403; 6,247,008; 6,578,027; 6,490,593; and 6,289,352. The Business Objects logo, the Business Objects tagline, BusinessObjects, BusinessObjects Broadcast Agent, BusinessQuery, Crystal Analysis, Crystal Analysis Holos, Crystal Applications, Crystal Enterprise, Crystal Info, Crystal Reports, Rapid Mart, and WebIntelligence are trademarks or registered trademarks of Business Objects SA in the United States and/or other countries. Various product and service names referenced herein may be trademarks of Business Objects SA. All other company, product, or brand names mentioned herein, may be the trademarks of their respective owners. Specifications subject to change without notice. Not responsible for errors or omissions. Copyright © 2004 Business Objects SA. All rights reserved.