

# Crystal Reports XI

## Java Beans Connectivity

---

### Overview

Crystal Reports XI Developer Edition has the capability of using Java Beans as a data source. This technical brief discusses how to construct a Crystal Reports-compatible Java Bean. Specifically, this document demonstrates how to create a Java Bean that returns a Java resultset containing data from the Xtreme Sample Database 11 that is installed with Crystal Reports. This document then discusses how to configure Crystal Reports to connect to and report off a Java Bean.

### Contents

<b>INTRODUCTION .....</b>	<b>2</b>
<b>CREATING A JAVA BEAN .....</b>	<b>2</b>
<i>Sample Java Bean.....</i>	<i>2</i>
<b>CHECKING FOR THE JAVA BEANS CONNECTIVITY FOLDER .....</b>	<b>4</b>
<b>CONFIGURING CRYSTAL REPORTS FOR JAVA BEANS CONNECTIVITY.....</b>	<b>5</b>
<i>Setting the path to the JDK.....</i>	<i>5</i>
<i>Setting the Java Beans classpath.....</i>	<i>5</i>
<b>CREATING REPORTS BASED ON JAVA BEANS.....</b>	<b>6</b>
<b>DEPLOYING REPORTS BASED ON JAVA BEANS.....</b>	<b>7</b>
<i>Java Reporting Component.....</i>	<i>8</i>
<i>Crystal Reports Server and BusinessObjects Enterprise.....</i>	<i>8</i>
<i>Report Designer Component and Crystal Reports for .NET.....</i>	<i>8</i>
<b>FINDING MORE INFORMATION .....</b>	<b>9</b>
<i>Knowledge base articles .....</i>	<i>9</i>
<i>Whitepapers .....</i>	<i>9</i>
<i>Utilities .....</i>	<i>9</i>

## Introduction

Java Beans connectivity was first introduced in Crystal Reports 9 Advanced Edition to allow any compiled Java class that has a public method returning a `java.sql.ResultSet` object to be used as a data source.

Reports based on Java Beans data sources can be processed using the Java Reporting Component (JRC), Crystal Reports Server, and BusinessObjects Enterprise.

**NOTE**

The OEM builds of Crystal Reports for Borland JBuilder, BEA WebLogic, and IBM Rational Application Developer do not have the ability to report off Java Beans.

## Creating a Java Bean

### Sample Java Bean

To create this sample Java Bean source file, complete these steps (these steps are not specific to Crystal Reports XI):

1. Create a text file called `CRSampleDataSourceBean.java` and copy the following code into this file:

```
import java.sql.*;

/*
 * <p>Title: Java Bean Data Source Sample</p>
 * <p>Description: Sample to demonstrate how to create a Java
 * Bean report data source with Crystal Reports XI</p>
 * <p>Copyright: Copyright (c) 2005</p>
 * <p>Company: Business Objects</p>
 * For Crystal Reports XI
 */

/*
 * For information on how to use JDBC and ResultSets see Sun's
 * tutorials and guides at
 * <a href="http://java.sun.com/products/jdbc/learning.html">.
 */

public class CRSampleDataSourceBean {
    private ResultSet resultSet = null;
    private Connection connection = null;
    private String connectionURL = "jdbc:odbc:Xtreme Sample
    Database 11";
    private String databaseClass =
        "sun.jdbc.odbc.JdbcOdbcDriver";
}
```

```
private String query = "SELECT * FROM CUSTOMER";

public CRSampleDataSourceBean() {
    try {
        // Ensure database class exists
        Class.forName(databaseClass);

        // Create a connection
        connection = DriverManager.getConnection(connectionURL,
            "", "");
    } catch (ClassNotFoundException ex) {
        System.out.println("Ensure that database driver class is
            installed.");
        ex.printStackTrace();
    } catch (SQLException ex) {
        System.out.println("SQL Exception #" + ex.getErrorCode() +
            " : " + ex.getLocalizedMessage());
        ex.printStackTrace();
    }
}

/*
 * Any function that returns a java.sql.ResultSet within
 * the Java class can be used in Crystal Reports to return
 * report data. The function can also be parameterized to
 * 'feed' in specific values to the function
 * when querying.
 */
public ResultSet getResultSet() throws SQLException {
    // Create an SQL statement to execute
    Statement statement =
        connection.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

    // Execute the select statement to populate the ResultSet
    resultSet = statement.executeQuery(query);
    return resultSet;
}
}
```

2. Save CRSampleDataSourceBean.java to the location where the Java Bean source files are located.
3. Compile the Java Bean source file.
4. Place the compiled .class file into a folder that is in the Java Bean's classpath. Crystal Reports searches this folder for the Java Bean Data Provider class files.

For information on setting up the Java Beans classpath, refer to the section [Setting the Java Beans classpath](#).

## Checking for the Java Beans connectivity folder

To verify whether your installation of Crystal Reports is properly set up to report off Java Beans, complete these steps to ensure that the **Java Beans Connectivity** folder exists:

1. In the Crystal Reports Designer, go to **File > New > Blank Report**.
2. In the **Database Expert** dialog box, expand the **Create New Connection** folder, and then search for the **Java Beans Connectivity** folder. (Figure 1)

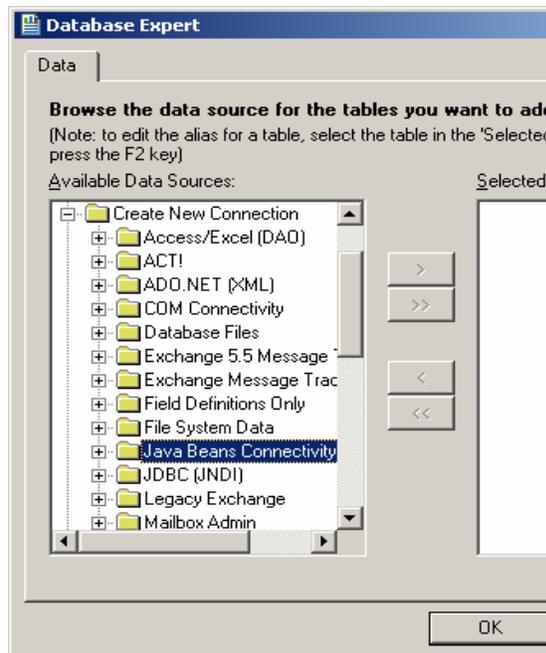


Figure 1

3. If the Java Beans Connectivity folder does not exist, go to **Add/Remove Programs** and click **Crystal Reports 11**.

4. Click the **Change** button then click the **Add/Remove** button to install additional components.
5. Expand **Data Access** then click **Java Data Provider**.
6. Click **Entire feature will be installed on local hard drive**.
7. Complete the remaining installation steps.

<b>NOTE</b>	<p>When you expand the Java Beans Connectivity folder, the following error message may appear:</p> <p>"Invalid Argument Provided. Details: Please Choose a Java Bean Class"</p> <p>This error message appears if Crystal Reports is not configured correctly. Configuration information is covered in the following section.</p>
-------------	--

## Configuring Crystal Reports for Java Beans connectivity

To configure Crystal Reports 9 and 10 for Java Beans connectivity, you needed to modify registry entries. In Crystal Reports XI, this is done by modifying the CRconfig.xml file that is located at the following location:

c:\Program Files\Common Files\Business Objects\3.0\java\

<b>NOTE</b>	<p>The CRconfig.xml file is also used for configuring JDBC and XML connectivity, as well as using Java-based User-Defined Function Libraries (UFLs).</p>
-------------	--

### Setting the path to the JDK

In the CRconfig.xml file, the <JavaDir> element contains the full path of an existing Java Runtime Environment (JRE) or J2SE Development Kit (JDK). If a JRE or JDK was detected by the Crystal Reports installation, this value will already be set.

If it is not set, download and install JDK 1.4 from <http://java.sun.com> and then set the value of <JavaDir> to the location of the file java.exe. For example:

```
<JavaDir>C:\Program Files\Java\JRE\bin</JavaDir>
```

### Setting the Java Beans classpath

In the CRconfig.xml file, the <JavaBeansClassPath> element must contain the locations of all the .class files that are to be used as Java Data Providers in Crystal Reports.

If the Java Bean class files are not jarred, set <JavaBeansClassPath> to the location of the .class file(s). For example:

```
<JavaBeans>
  <CacheRowsetSize>100</CacheRowsetSize>
  <JavaBeansClassPath>C:\jbean</JavaBeansClassPath>
</JavaBeans>
```

If the Java Bean class files are jarred, set <JavaBeansClassPath> to the full path to the .jar file(s). For example:

```
<JavaBeans>
  <CacheRowsetSize>100</CacheRowsetSize>
  <JavaBeansClassPath>C:\myjbean.jar</JavaBeansClassPath>
</JavaBeans>
```

## Creating reports based on Java Beans

You are now ready to create a report off your Java Bean data source. Complete the following steps:

1. In the Crystal Reports Designer, go to **File > New > Blank Report**. The **Database Expert** dialog box appears.
2. Expand the **Create New Connection** folder and then expand the **Java Beans Connectivity** folder. The **Java Beans Connectivity** dialog box appears displaying a list of available Java classes (for example, see Figure 3).

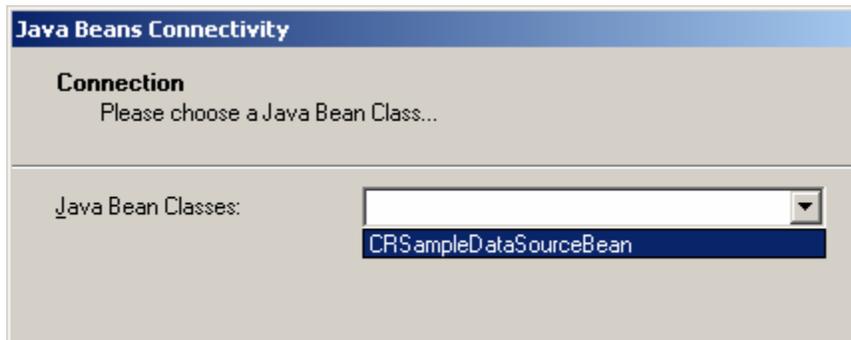
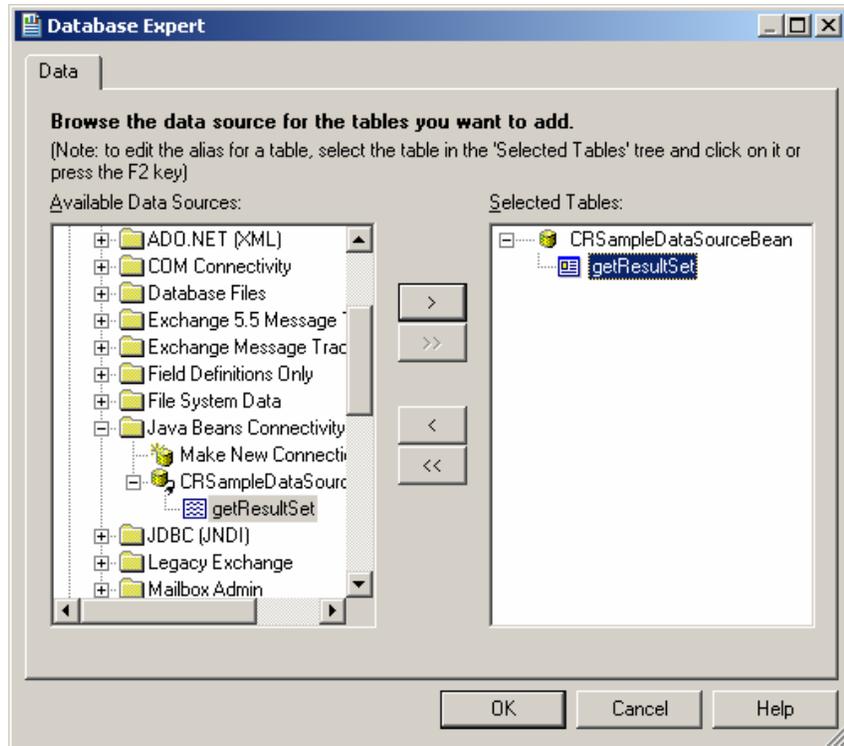


Figure 3

3. Click a Java Bean in the **Java Bean Classes** list, and then click the **Finish** button. The methods defined in the Java Bean that return a `java.sql.ResultSet` object are listed as tables under the **Java Beans Connectivity** folder (for example, see Figure 4).



*Figure 4*

4. Click one of the methods to use as a table in the report and then click **OK**.

You are now connected to the Java Bean and can create a report in the Crystal Reports Designer.

## Deploying reports based on Java Beans

Reports based on Java Beans can be deployed to embedded reporting applications using the Java Reporting Component (JRC), Crystal Reports Server, and BusinessObjects Enterprise. In addition, applications built around the Crystal Reports for Visual Studio .NET and the Report Designer Component (RDC) technologies can also be used to process reports with Java Bean data sources.

The following sections describe how to configure each type of reporting technology for processing and distributing reports based on Java Bean data sources.

## Java Reporting Component

When deploying reports based on Java Beans for processing using the JRC, insert the Java Bean .class file(s) into the application's /WEB-INF/class folder. Alternatively, jar the .class file(s) and then place the JAR file in the web application's /WEB-INF/lib folder.

Ensure that the necessary JAR files from the C:\Program Files\Common Files\Business Objects\3.0\java\lib\external folder, as well as the required JRC engine JAR files have also been included in the web application's /WEB-INF/lib folder.

## Crystal Reports Server and BusinessObjects Enterprise

When deploying reports based on Java Beans for processing using Crystal Reports Server or BusinessObjects Enterprise, complete the following steps:

1. Place the Crystal Reports CRconfig.xml file in the C:\Program Files\Common Files\Business Objects\3.0\java folder.
2. If you are using BusinessObjects Enterprise, set the value of the <JavaDir> and <JavaBeansClassPath> elements on the computers running the Crystal Report Page Server, Crystal Report Job Server, and Crystal Report Application Server services. If you are using Crystal Reports Server, these services would be running on the same computer.
3. The Java Bean .class file must be local to all these processing servers, otherwise the error message "Unknown Query Engine Error" will appear.

## Report Designer Component and Crystal Reports for Visual Studio .NET

To process reports based on Java Beans using the Report Designer Component (RDC) or Crystal Reports for .NET, complete the steps in the section [Configuring Crystal Reports for Java Beans Connectivity](#) for the Crystal Reports Designer that is installed on the same computer as Crystal Reports for .NET or the RDC.

## Finding More Information

### Knowledge base articles

Search for the following related knowledge base articles on our [support site](#):

- c2017726 - Error: "Failed to retrieve data from database" passing parameters to a JavaBean
- c2011506 - Configuring Crystal Reports 10 to use Java Bean classes as data sources
- c2011653 - Limitations when using crdb\_javabeans.dll to report on Java Beans
- c2011830 - Troubleshooting Err Msg: "Unknown Query Engine Error" when using Java Bean
- c2011834 - Report appears blank when using Java Bean as data source in CR 10
- c2012139 - Err Msg: "Query Engine Error: Failed to find the Java Class: 'JavaClassName'"
- c2013206 - Err Msg: "Query engine error..." when using Java Beans Connectivity in CR 9
- c2015714 - How to debug a connection to a Java Bean by retrieving the JVM error messages
- c2015416 - How to pass a parameter value to a Java Bean data source

### Whitepapers

Understanding CRConfig.xml for Crystal Reports XI:

[http://support.businessobjects.com/communityCS/TechnicalPapers/crxi\\_crconfigxml.pdf.asp](http://support.businessobjects.com/communityCS/TechnicalPapers/crxi_crconfigxml.pdf.asp)

### Utilities

SQLCon32 utility to troubleshoot database connections:

<http://support.businessobjects.com/communityCS/FilesAndUpdates/sqlcon32.zip>

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

No part of the computer software or this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from Business Objects.

The information in this document is subject to change without notice. Business Objects does not warrant that this document is error free.

This software and documentation is commercial computer software under Federal Acquisition regulations, and is provided only under the Restricted Rights of the Federal Acquisition Regulations applicable to commercial computer software provided at private expense. The use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013.

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 © 2005 Business Objects SA. All rights reserved.