Getting Started with JasperReport

- June 23, 2014
- By Manoj Debnath
- Bio »
- Send Email »
- More Articles »

JasperReport is a popular reporting tool used by the Java programmer. Interestingly, what started as Teodor Danciu's (creator of JasperReport library, 2001) need for an inexpensive reporting solution, has today bloomed into a full-fledged reporting library. This facilitated the need for a simple, inexpensive yet feature rich tool to add reporting capabilities in a Java application.

Much like other Java libraries, JasperReport is an API to facilitate reporting output in a variety of formats, be it, PDF, XML, HTML, XLS, etc. Further this library is not limited to adding reporting capabilities to web-based applications only; it can also be used to generate reports from desktop and console applications as well. In this article we shall see how to go hands on rather than delving into its architectural details.

Fundamentals

Though it is a library like any other in Java, there are a few points to be noted while working with JasperReport.

1) Obviously, Jasper Library needs to be added to our project and some sort of layout has to be generated before we are able to start reporting from Java code.

2) Jasper's reporting layout design is nothing but an XML file with the extension <filename>.jrxml.

3) This JRXML file is to be compiled to create <filename>.jasper. JRXML file can be compiled on the fly, dynamically from our Java code or we can use iReport or JasperStudio to visually design the JRXML file and then compile to create a Jasper file.

4) Once compiled and <filename>.jasper are created, we are done and can feed data into the report from the Java code.

Note: In this article, we shall be using NetBeans 8.0 and iReport5.5.0 plugin. This plugin was actually meant for Netbeans 7.4 but works fine for NetBeans 8.0 as well.

So What We Need

Related Articles

- Get Ready for Java 7 with the Free JDK 7 Reference Card
- Java 7 and 8 Begin to Take Shape: What's In, What's Out?

JasperReport heavily depends on other libraries such as Apache log4j, JFreeChart, Apache Commons, etc. As a result dependencies are a little sensitive regarding versions mismatch, especially when its files are downloaded and integrated from scattered locations. The good news is that JasperReport project contains everything we need. Once you download the zip file you need not worry. The details about the library and other needs, to go hands on in accordance to this article are as follows. The versions given here are the latest, at least the time of writing this article.

1. JDK 8.0
2. Netbeans 8.0
3. iReport5.5.0 Netbeans 7.4 plugin
4. JaspeReports-5.6.0-project

Installation and Configuration

- Installing JDK and Netbeans is pretty straightforward, so I'm not going into the details.
- Once the iReport5.5.0 plugin is downloaded, unzip it in any location. Open Netbeans, select **Plugins** from the **Tools** menu. From the **Download** tab select **Add Plugins...** and open all **.nbm* files.

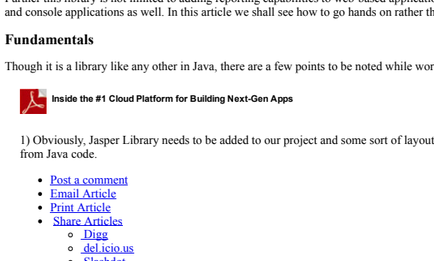

Figure 1: Adding plugins

- To create a JasperReport user defined library, select **Libraries** from the **Tools** menu. Click on the **New Library...** button at the bottom left corner of the window, give a **Library Name → OK**. Then click on **Add Jar/Folder...** and navigate to the location where you have unzipped JasperReport5.6.0 library and include everything from the **lib** and **dist**folders.

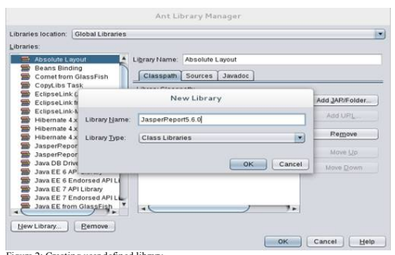

Figure 2: Creating user defined library

*Note: There is a huge list of jar files in the **lib** and **dist** folders of the Jasper library. Every project may not need all of the jar files, but it doesn't hurt to create a reusable library in NetBeans with all the jars so that we can include it in our Java project without bothering about what jar to include and what not to include. This is especially helpful for beginners where identifying a particular jar may be difficult at times.*

We are now ready to create a Java project with Jasper reporting.

Starting a Project

Creating a Simple Application with Jasper Reporting

1. File --> New Project...
2. Select **Java** from **Categories** and **Java Application** from **Projects**. Then **Next.**
3. Give the name of the application and click **Finish**.
4. Add JasperLibrary5.6.0 to the project by right clicking **Libraries** from the **Project** pane and **Add Library** and then **import**.
5. Open the **Files** pane (**Window --> Files**). Right click on the project root to create a folder named *reports.*
6. Right click on the *reports* folder then select **New → Empty report...**
7. Give the name of the report '*report1*' and **Finish.**

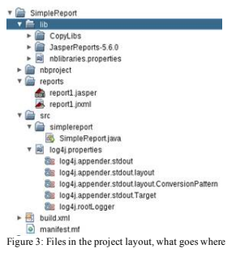

Figure 3: Files in the project layout, what goes where

Designing a Report

Open report1.jrxml in **Designer** view and use **Report Inspector** to design the report as in figure 4.

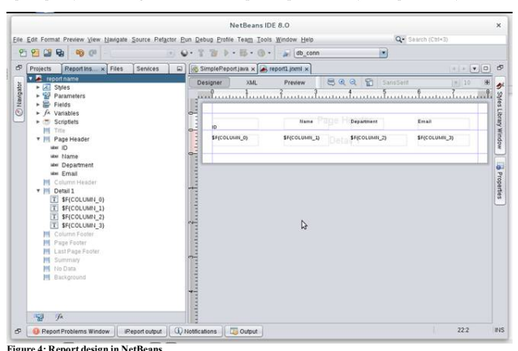

Figure 4: Report design in NetBeans

The corresponding XML details of the *report1.jrxml* are shown in **Listing 1.**

Listing 1: report1.jrxml

```
<?xml version="1.0" encoding="UTF-8"?>
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports http://jasperreports.sourceforge.net/xsd/jasperreport.xsd" name="AircraftReportColumnIndex" pageWidth="595" pageHeight="842" columnWidth="555" leftMargin="20" rightMargin="20" topMargin="30" bottomMargin="30" uuid="6e843fe2-ce72-4257-a4c3-1df03728dc95">
    <property name="ireport.zoom" value="1.0"/>
    <property name="ireport.x" value="0"/>
    <property name="ireport.y" value="0"/>
    <field name="COLUMN_0" class="java.lang.String"/>
```

```
    <field name="COLUMN_1" class="java.lang.String"/>
    <field name="COLUMN_2" class="java.lang.String"/>
    <field name="COLUMN_3" class="java.lang.String"/>
    <pageHeader>
        <band height="30" splitType="Stretch">
            <staticText>
                <reportElement x="0" y="0" width="69" height="24" uuid="012424cf-712d-4e84-9906-776e1850b85a"/>
                <textElement verticalAlignment="Bottom">
                    <font size="10" isBold="false"/>
                </textElement>
                <text><![CDATA[ID]]></text>
            </staticText>
            <staticText>
                <reportElement x="140" y="0" width="94" height="24" uuid="724d23ca-6ad1-4be5-bae1-77c07dd31ba0"/>
                <textElement textAlignment="Center"/>
                <text><![CDATA[Name]]></text>
            </staticText>
            <staticText>
                <reportElement x="280" y="0" width="69" height="24" uuid="1e85a3f6-ba9d-47a7-8f25-cf37f5b4448d"/>
                <text><![CDATA[Department]]></text>
            </staticText>
            <staticText>
                <reportElement x="420" y="0" width="108" height="24" uuid="044a8958-4960-4fa3-9cd6-c594595c521a"/>
                <text><![CDATA[Email]]></text>
            </staticText>
        </band>
    </pageHeader>
    <detail>
        <band height="30" splitType="Stretch">
            <textField>
                <reportElement x="0" y="0" width="69" height="24" uuid="d844cada-1aa4-4208-9fc1-dcdf62a72235"/>
                <textFieldExpression><![CDATA[$F{COLUMN_0}]]></textFieldExpression>
            </textField>
            <textField>
                <reportElement x="140" y="0" width="94" height="24" uuid="14399970-e399-41e0-b6f9-1218079fd56c"/>
                <textFieldExpression><![CDATA[$F{COLUMN_1}]]></textFieldExpression>
            </textField>
            <textField>
                <reportElement x="280" y="0" width="69" height="24" uuid="b5b0fe03-9b8f-48c6-ba51-c218427028f6"/>
                <textFieldExpression><![CDATA[$F{COLUMN_2}]]></textFieldExpression>
            </textField>
            <textField>
                <reportElement x="420" y="0" width="108" height="24" uuid="c3094477-bb5e-4d5c-a440-8d7c7f2a1d3e"/>
                <textFieldExpression><![CDATA[$F{COLUMN_3}]]></textFieldExpression>
            </textField>
        </band>
    </detail>
</jasperReport>
```

Add log4j.properties

Create a log4j.properties file inside the **src** folder. Without this, file compiler would complain such as:

```
log4j:WARN No appenders could be found for logger (net.sf.jasperreports.engine.xml.JRXmlDigesterFactory).
log4j:WARN Please initialize the log4j system properly.
```

Listing 2: log4j.properties

```
log4j.rootLogger=ERROR, stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d %5p %c:%L - %m%n
```

Create Java Class

In the Java class we are creating a report from a model of table data. This is our basic data source in this project. We are also using *JasperCompileManager* to compile the jrxml file and produce .jasper file dynamically through Java code. This Jasper file is passed to the *JasperPrint* class along with table data from *TableModel*. JasperPrint object acts as a container of data from the data source. The report now is ready.

To view the report we then pass the initialized *JasperPrint* object to create a new *JasperView* object. *JasperView* extends Swing *JFrame* class and hence provides a GUI frame to display the report as in figure 5.

Listing 3: SimpleReport.java

```
//...import  statements

public class SimpleReport {
    DefaultTableModel tableModel;

    public SimpleReport() {
        JasperPrint jasperPrint = null;
        TableModelData();
        try {
            JasperCompileManager.compileReportToFile("reports/report1.jrxml");
            jasperPrint = JasperFillManager.fillReport("reports/report1.jasper", new HashMap(),
                    new JRTableModelDataSource(tableModel));
            JasperViewer jasperViewer = new JasperViewer(jasperPrint);
            jasperViewer.setVisible(true);
        } catch (JRException ex) {
            ex.printStackTrace();
        }

    }

    private void TableModelData() {
        String[] columnNames = {"Id", "Name", "Department", "Email"};
        String[][] data = {
            {"111", "G Conger", " Orthopaedic", "jim@wheremail.com"},
            {"222", "A Date", "ENT", "adate@somemail.com"},
            {"333", "R Linz", "Paedriatics", "rlinz@heremail.com"},
            {"444", "V Sethi", "Nephrology", "vsethi@whomail.com"},
            {"555", "K Rao", "Orthopaedics", "krao@whatmail.com"},
            {"666", "V Santana", "Nephrology", "vsan@whenmail.com"},
            {"777", "J Pollock", "Nephrology", "jpol@domail.com"},
            {"888", "H David", "Nephrology", "hdavid@donemail.com"},
            {"999", "P Patel", "Nephrology", "ppatel@gomail.com"},
            {"101", "C Comer", "Nephrology", "ccomer@whymail.com"}
        };
        tableModel = new DefaultTableModel(data, columnNames);
    }

public static void main(String[] args) {
        new SimpleReport();
    }

}
```

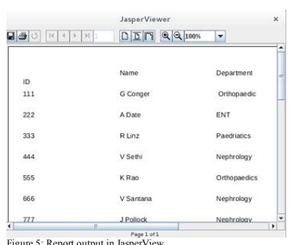

Figure 5: Report output in JasperView

Conclusion

The above example is very simple yet effective method of getting hands on in one's first Jasper report projects. The tags used in JRXML though looks like a simple XML yet needs further explanation; readers will get a thorough idea if JasperReport documentation is followed. In future articles we will delve a little deeper into how we can use JasperReport further to meet our reporting needs.

Tags: Java, API, report, library

15 Comments (click to add your comment)

Comment Page: 1
2
By Yan May 03 2016 23:19 PDT

Hello, Can you give import Statements?

Reply to this comment By amokrane April 23 2016 03:45 PDT

where find an example about using List component in report ? nowehre on the net! my problme is : how to passe a jList items from java code to a List component into report? help plaise!!!

Reply to this comment By Robs April 08 2016 10:54 PDT

When I run the program It first shows this error Error: Unknown Host: jasperreports.sourceforge.net URL Resource Then it goes on to give this output in the report window net.sf.jasperreports.engine.design.JRValidationException: Report design not valid : 1. Field not found : COLUMN_0 2. Field not found : COLUMN_1 3. Field not found : COLUMN_2 4. Field not found : COLUMN_3 at net.sf.jasperreports.engine.design.JRAbstractCompiler.verifyDesign(JRAbstractCompiler.java:271) BUILD SUCCESSFUL (total time: 31 seconds)

Reply to this comment By Nirmal October 30 2015 06:07 PDT

You save my ass man.!!! @Shiva: That was really helpful.

Reply to this comment By anjali October 28 2015 05:52 PDT

Please remove all the uuid tag for removing FileNotFoundException

Reply to this comment By nilar October 25 2015 04:15 PDT

I have a problem, I used your tutorials that run perfectly in netbeans IDE after that I made an executable jar file. When I run executable jar file "net.sf.jasperreports.engine.JRException: java.io.FileNotFoundException: reports\reprot1.jrxml(The system cannot find the path specified)" error occur and any report not appear. If you receive this comment please reply sir! I need your help?

Reply to this comment By rahul October 23 2015 11:47 PDT

I have Jasper report project 6.1.1 Can i use this project library but you say use 5.6.1 plz suggest me solution fast

Reply to this comment By sravan September 22 2015 01:23 PDT

im getting following errors while creating a jasper report in my project. Buildfile: /sitrack/src/sitrackReports/build.xml clean: [delete] Deleting directory /sitrack/src/sitrackReports/build [delete] Deleting directory /sitrack/src/sitrackReports/dist [mkdir] Created dir: /sitrack/src/sitrackReports/build [mkdir] Created dir: /sitrack/src/sitrackReports/build/classes [mkdir] Created dir: /sitrack/src/sitrackReports/build/reports [mkdir] Created dir: /sitrack/src/sitrackReports/build/temp [mkdir] Created dir: /sitrack/src/sitrackReports/dist compileReports: [jrc] Compiling 3 report design files. [jrc] log4j:WARN No appenders could be found for logger (net.sf.jasperreports.engine.xml.JRXmlDigesterFactory). [jrc] log4j:WARN Please initialize the log4j system properly. [jrc] File : /sitrack/src/sitrackReports/reports/materialInventory/MaterialTypeStockReport.jrxml ... OK. [jrc] File : /sitrack/src/sitrackReports/reports/mySecondReport/MySecondReport.jrxml ... FAILED. [jrc] Error compiling report design : /sitrack/src/sitrackReports/reports/mySecondReport/MySecondReport.jrxml [jrc] net.sf.jasperreports.engine.design.

Reply to this comment By shiva August 18 2015 22:16 PDT

If you get the error (File not found - jasper file) try the following codes.... public SimpleReport() { JasperPrint jasperPrint = null; net.sf.jasperreports.engine.JasperReport x = null; TableModelData(); try { x = JasperCompileManager.compileReport("reports/report1.jrxml"); jasperPrint = JasperFillManager.fillReport(x, new HashMap(), new JRTableModelDataSource(tableModel)); JasperViewer jasperViewer = new JasperViewer(jasperPrint); jasperViewer.setVisible(true); } catch (JRException ex) { ex.printStackTrace(); } } All the best.....

Reply to this comment By amar June 10 2015 00:42 PDT

I need the codes to fetch unicode to hindi into pdf file hw to convert it..plzz help. !!!!

Reply to this comment **More comments ... Page:** 1
2

Comment and Contribute

Your name/nickname

Your email

Subject

(Maximum characters: 1200). You have 1200 characters left.

Submit Your Comment

Enterprise Development Update

Don't miss an article. Subscribe to our newsletter below.
Enter Email Address SIGN UP

Most Popular Developer Stories

- Today
- This Week
- All-Time

- 1 Using JDBC with MySQL. Getting Started
- 2 An Introduction to Java Annotations
- 3 MIDP Programming with J2ME
- 4 An Introduction to JSP Standard Template Library (JSTL)
- 5 Debugging a Java Program with Eclipse

- 1 Using JDBC with MySQL. Getting Started
- 2 An Introduction to Java Annotations
- 3 An Introduction to JSP Standard Template Library (JSTL)
- 4 MIDP Programming with J2ME
- 5 Debugging a Java Program with Eclipse

- 1 Using JDBC with MySQL. Getting Started
- 2 An Introduction to Java Annotations
- 3 An Introduction to JSP Standard Template Library (JSTL)
- 4 MIDP Programming with J2ME
- 5 Debugging a Java Program with Eclipse

Most Commented On

- This Week
- This Month

- All-Time

- 1 10 Experimental PHP Projects Pushing the Envelope
- 2 Day 1: Learning the Basics of PL/SQL
- 3 C# Tip: Placing Your C# Application in the System Tray
- 4 Logical Versus Physical Database Modeling
- 5 Is Ubuntu Contributing as Much as It Should to Free Software Projects?

- 1 Day 1: Learning the Basics of PL/SQL
- 2 The 5 Developer Certifications You'll Wish You Had in 2015
- 3 10 Experimental PHP Projects Pushing the Envelope
- 4 An Introduction to Struts
- 5 Inside Facebook's Open Source Infrastructure

- 1 Creating Use Case Diagrams
- 2 Day 1: Learning the Basics of PL/SQL
- 3 C# Tip: Placing Your C# Application in the System Tray
- 4 Using ASP.NET To Send Email
- 5 Using JDBC with MySQL, Getting Started

Thanks for your registration, follow us on our social networks to keep up-to-date