CHAPTER 1

Getting Started

This chapter is for those who have never used JasperReports. If you have already installed JasperReports and used the samples, skip to Chapter 2.

Installing JasperReports

JasperReports is a content-rendering library, not a standalone application. It cannot run on its own and must be embedded in another client- or server-side Java application. JasperReports is a pure Java library and can be used on any platform that supports Java. Being a library, JasperReports is completely agnostic about the environment in which it is used for generating reports.

All JasperReports functionality is concentrated in a single JAR file, `jasperreports-x.x.x.jar`, available for download at the following URL:

```
http://sourceforge.net/project/showfiles.php?group_id=36382&package_id=28579
```

Even though all its reporting functionality is available in this single JAR file, JasperReports relies on other third-party libraries for related required functionality like XML parsing, logging, and PDF and XLS generation.

Because JasperReports is a library and cannot run on its own, you do not really install it. “Installing” JasperReports simply means downloading its JAR file and putting it into the classpath of your application along with the other required JAR files.

Requirements

JasperReports handles only reporting. It relies on third-party libraries and APIs to perform needed functionality like XML parsing, database connectivity, PDF or XLS output, and so on.

This section contains all the libraries that JasperReports may require, depending on the functionality required by JasperReports’s parent application.

The two types of requirements for using JasperReports are the following:

- Absolute requirements, needed regardless of the module of JasperReports that is actually used.
- Optional requirements, needed only for a specific JasperReports function. (If a certain function of JasperReports is not used by the parent application, then the required libraries needed by that module can be skipped at deployment time.)

The following list details the requirements for using JasperReports:

- Java Virtual Machine (JVM), JRE 1.3 or higher
- One of the following for report compilation, depending on the report compiler used:
  - Eclipse JDT Java compiler (`www.eclipse.org/jdt/index.php`)
  - JDK 1.3 or higher
  - Jikes (`http://jikes.sourceforge.net`)
  - Groovy (`http://groovy.codehaus.org`)
  - BeanShell (`www.beanshell.org`)
- JAXP 1.1 XML Parser
- Jakarta Commons Javaflow, sandbox version (`http://jakarta.apache.org/commons/sandbox/javaflow`)
- Jakarta Commons Digester component, version 1.7 or later (`http://jakarta.apache.org/commons/digester`)
- Jakarta Commons BeanUtils component, version 1.4 or later (`http://jakarta.apache.org/commons/beanutils`)
- Jakarta Commons Collections component, version 2.1 or later (`http://jakarta.apache.org/commons/collections`)
- Jakarta Commons Logging component, version 1.0 or later (`http://jakarta.apache.org/commons/logging`)
- JDBC 2.0 driver
- iText (free Java PDF library by Bruno Lowagie and Paulo Soares), version 1.01 or later (`www.lowagie.com/iText`)
- The following APIs for XLS:
  - Jakarta POI, version 2.0 or later (`http://jakarta.apache.org/poi`)
  - JExcelApi, version 2.6 or later (`http://jexcelapi.sourceforge.net`)
- JFreeChart (free Java chart library), version 1.0.0 or later (`www.jfree.org/jfreechart`)

X11/Headless Java

JasperReports relies on AWT rendering when generating reports, so it might not work if you are using it in a server environment running UNIX/Linux without graphics support.

The application might raise errors such as “Can’t connect to X11 window server using ‘:0.0’.”

To solve this problem for JVM releases prior to 1.4, provide a pseudo–X server to emulate a display environment. Following are some of these emulators:

- X Virtual Frame Buffer (Xvfb)
- Pure Java AWT (PJA)
- Virtual Network Computing (VNC)

The preferred solution for JRE 1.4 or higher is to use the new headless AWT toolkit. This new feature allows you to use the J2SE API in a server-side Java application without a GUI environment.

To specify the headless environment when using the Sun Microsystems reference implementation, run your application with this property:

```
-Djava.awt.headless=true
```

Building the Source Files and Running the Samples

The best way to start working with JasperReports is to download the full project package from the following SourceForge.net location:

```
http://sourceforge.net/project/showfiles.php?group_id=36382&package_id=28579
```

The `jasperreports-x.x.x-project.zip` file available at this location contains all the source files, required libraries, and freely available documentation, as well as a complete set of sample applications and reports.

Download the archive and extract its contents to the directory of your choice on your local machine. You'll be able to see JasperReports in action without having to create a Java application to embed JasperReports in.

Ant Build Tool

Before using the JasperReports distribution files and samples, install the Ant tool on your machine.

JasperReports relies heavily on the Ant build tool from the Apache Foundation (`http://ant.apache.org`) to compile the source files, build the distribution files, generate the Javadoc documentation, and run the samples. The Ant build tool will make working with the JasperReports library easier. Please refer to the Ant documentation for installation instructions.

Building the Project from Source Files

Once you have installed Ant, you can compile the source files, generate the Javadoc API documentation, or build the distribution JAR files. To do this, execute the Ant tasks declared in the `build.xml` file found in the root directory of the project tree.

To see details of each available task, launch the `ant -p` command from the command prompt inside this root directory.

Running the Samples

The JasperReports distribution package comes with a complete set of sample applications and reports that show how each individual feature of the library can be used.

The samples are in the `/demo/samples` directory inside the project tree.

HSQLDB Demo Database

Some of the samples use data from an HSQLDB demo database supplied in the `/demo/hsqldb` directory of the JasperReports distribution (`www.hsqldb.org`).

Before running those samples, start the HSQLDB database by going to the `/demo/hsqldb` directory and launching `ant runServer` from the command prompt. To look into the database content using a simple SQL client tool, launch the HSQLDB Manager application by invoking `ant runManager` in the same directory after starting the database.

To test a particular sample in the `/demo/samples` directory, go to the corresponding sample subfolder and launch `ant -p` from the command line.

This displays a short description of what that sample demonstrates as well as a complete list of Ant tasks available for use on the sample's source files.

The following list gives the typical steps for running a sample:

1. Compile the sample's Java source files by calling `ant javac`.
2. Compile the JRXML report templates used by the sample application with `ant compile`.
3. Fill those report templates with data by calling `ant fill`.
4. View the result with `ant view`.

To export to other formats, simply use commands like `ant pdf` or `ant html`.

These samples are used throughout this guide to illustrate features of the library. Therefore make sure you can run the samples before reading the rest of this guide.