

Sample Code Walkthrough

Overview

This Windward Tutorial will walk through the basic sample code included with the Java Engine. This code reads in a template (.docx, .xlsx, or .pptx formats), connects to a database (MySQL, MS SQL, DB2 and XML) and produces an output file in PDF format

This tutorial is part of the Trebuchet app, included with the Windward Java Engine. It's purpose is to help you become familiar with the Java Engine, and launch you with a high velocity into the world of simple reporting!

Requirements

- Windward Reports Java Engine
 - Available for download at <http://www.windward.net/downloads.php>
 - Installation Tutorial
- Java JDK v1.4 or later
 - Available for download at <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
 - Installation Tutorial
- Database connector - dependent on the format of your database
 - DB2
 - MS SQL
 - MySQL

Tutorial

The simple application which takes a template and generates a report only requires 5 lines of code. By adding to these 5 lines of code, Windward engine users can extend functionality in any way imaginable that fits the scope of a Java application.

```
ProcessReportAPI report = new ProcessPdf(new FileInputStream("data/template.docx"),  
new FileOutputStream("out/report.pdf"));
```

Here we create the report, and we'll add stuff to it after we process it. Since we plan on creating a PDF document to store the report in, we'll instantiate this report as a ProcessPdf object. You are not limited in choice here; you can create a report in a variety of formats including HTML, Text, Excel/Word files, or you can have it sent straight to the printer.

The parameters are simple, an in-file for the template and an out-file for the report. The template can be a DOCX, XLSX or PPTX file with properly inserted Windward Tags.

```
report.processSetup();
```

This method call simply verifies that a report can be run, given the conditions. It is a required call but only needs to be called once per report.

This next line will be different depending on the format of your database. This line loads your data source, by creating a DataSourceProvider. The parameters form the connection string.

- For the three **sql types** (DB2, MS SQL and MySql) the parameters are as follows: new JdbcDataSource(driver classpath, url, user, pass);
- **Also for the three sql types:** The drivers are not included in the class path, so you will have to include them yourself. Instructions for how to do this in Eclipse and IntelliJ are included with the Trebuchet app.

XML	<pre>DataSourceProvider datasource = new Dom4jDataSource(new FileInputStream("data/data.xml"));</pre> <p>For XML, the parameter is quite easy. You just need an xml file. For this example, we have data.xml (which you can find in the project folder included in the Trebuchet app)</p>
DB2	<pre>DataSourceProvider datasource = new JdbcDataSource("com.ibm.db2.jcc.DB2Driver", "jdbc:db2://db2.windward.net:50000/SAMPLE", "demo", "demo");</pre> <p>The URL parameter is just a standard URL, db2.windward.net is the server, 50000 is the port number and SAMPLE is the database name.</p>
MS SQL	<pre>DataSourceProvider datasource = new JdbcDataSource("com.microsoft.sqlserver.jdbc.SQLServerDriver", "jdbc:sqlserver://mssql.windward.net;DatabaseName=Northwind", "demo", "demo");</pre> <p>The URL parameter is a little bit different for MS SQL, but it should still be pretty straightforward.</p>
MySql	<pre>DataSourceProvider datasource = new JdbcDataSource("com.mysql.jdbc.Driver", "jdbc:mysql://mysql.windward.net/sakila", "demo", "demo");</pre> <p>The URL parameter is, again, very straightforward.</p>

```
report.processData(datasource, "Orders");
```

This line does more than any other line here. Using the template and data source provided, it processes the report and stores it in the report file provided. The parameters are the data source, and the name of the database. This name isn't necessarily the same name the database may have on its server; templates may use multiple data sources, so this is the name the tags in the template use to refer to separate data sources.

```
report.processComplete();
```

And with that, you are done. This finishes up the process and saves the final file which you should take a look at to see how well Windward works!