

# The JFreeReport Class Library

Version 0.8.3

## An Introduction

Written by David Gilbert

May 27, 2003

© 2000-2003, Object Refinery Limited. All rights reserved.

*Everyone is permitted to copy and distribute verbatim copies of this document,  
but changing it is not allowed.*

### **IMPORTANT NOTICE:**

**If you choose to use this document you do so entirely at your own  
risk.**

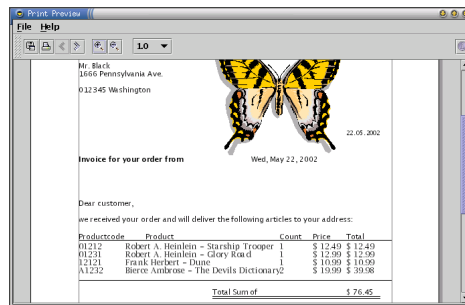
## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	What is JFreeReport?	2
1.2	The JFreeReport Project	2
1.3	This Document	2
1.4	Disclaimer	2
1.5	Acknowledgements	3
1.6	Comments and Suggestions	3
<b>2</b>	<b>Downloading and Installing JFreeReport</b>	<b>4</b>
2.1	Introduction	4
2.2	Download	4
2.3	Installation	4
2.4	Running the Demonstration Application	5
2.5	Compiling the Source	6
2.6	Generating the Javadoc Documentation	6
<b>3</b>	<b>Getting Started</b>	<b>7</b>
3.1	Introduction	7
3.2	The Basic Steps	7
3.3	Sample Application: Hello World	8
3.4	Sample Application: SwingIconsDemo	10
3.5	The Report Template File	12
3.6	Previewing the Report	29
<b>A</b>	<b>The GNU Lesser General Public Licence</b>	<b>31</b>
A.1	Introduction	31
A.2	The Licence	31

# 1 Introduction

## 1.1 What is JFreeReport?

JFreeReport is a free Java™ class library for generating reports. JFreeReport data is sourced via Swing's `TableModel` interface,<sup>1</sup> and formatted according to an XML-based report template file. Reports can be previewed on screen, sent to the printer or saved in various formats including PDF, HTML, Excel, CSV and plain text format.



JFreeReport is free under the terms of the GNU Lesser General Public Licence (LGPL)—see Appendix A for details.

## 1.2 The JFreeReport Project

JFreeReport's official home page is:

<http://www.jfree.org/jfreereport/index.html>

Please visit this site for news, updates, and access to the JFreeReport forum.

Thomas Morgner is the JFreeReport Project Leader. Development activities are coordinated through the project site hosted at SourceForge:

<http://sourceforge.net/projects/jfreereport/>

New developers are always welcome! If you would like to help out, please visit the project site and sign up.

## 1.3 This Document

This document has been written (by David Gilbert) for version 0.8.3 of JFreeReport.

## 1.4 Disclaimer

Please note that I cannot guarantee that this document is error free. You must use this document *at your own risk* or *not use it at all*.

<sup>1</sup>Used to obtain data for tables created with Swing's `JTable` class.

## 1.5 Acknowledgements

The latest release of JFreeReport is the end result of the hard work and dedication of the following people: Thomas Morgner, David Gilbert, Piotr Bzdyl, Heiko Evermann, Patrice Rolland, Joerg Schoemer and others.<sup>2</sup>

JFreeReport also leverages the work of other developers, such as Bruno Lowagie and Paulo Soares (iText developers) and the Apache POI project team. A big thank-you to them for sharing their excellent work.

Finally, thanks to all the developers that have provided feedback (bug reports and feature requests) concerning JFreeReport.

## 1.6 Comments and Suggestions

If you have any comments or suggestions regarding this document, please post a message in the JFreeReport discussion forum.

---

<sup>2</sup>Please e-mail me ([david.gilbert@object-refinery.com](mailto:david.gilbert@object-refinery.com)) if I've forgotten to add your name here!

## 2 Downloading and Installing JFreeReport

### 2.1 Introduction

This section contains instructions for:

- downloading and installing (unpacking) JFreeReport;
- running the demonstration applications;
- compiling<sup>3</sup> the JFreeReport source code;
- generating the Javadoc HTML documentation from the source code;

### 2.2 Download

You can download the latest version of JFreeReport from the JFreeReport project page on SourceForge:

<http://sourceforge.net/projects/jfreereport>

For version 0.8.3, you can choose between two download archives:

File:	Description:
<code>jfreereport-0.8.3.zip</code>	A <code>zip</code> file containing all the files required for JFreeReport (recommended for users of Microsoft Windows).
<code>jfreereport-0.8.3.tar.gz</code>	A <code>tar.gz</code> file containing all the files required for JFreeReport (recommended for users of Linux or Unix operating systems).

Both archives contain exactly the same source code, except that the end-of-line markers have been encoded using CR/LF in the `zip` archive, and LF only in the `tar.gz` archive.

Whichever archive you choose to download, you should save it in the directory where you want the `jfreereport-0.8.3` directory (created during unpacking—see the next section) to reside.

### 2.3 Installation

#### 2.3.1 Overview

Installing JFreeReport is simply a matter of unpacking the download archive. When you unpack the archive, a new directory (`jfreereport-0.8.3`) will be created at the same location in your filesystem as the archive file itself.

#### 2.3.2 Unpacking the ZIP Archive

If you chose to download the `jfreereport-0.8.3.zip` archive, you can “unzip” it using any ZIP utility. The `jar` utility included with Java will do the job—type the following command:

```
jar -xvf jfreereport-0.8.3.zip
```

<sup>3</sup>Recompilation is optional, because the precompiled runtime jar files are included in the download.

### 2.3.3 Unpacking the tar.gz Archive

If you chose to download the `jfreereport-0.8.3.tar.gz` archive, you can unpack it using the following command:

```
tar xvzf jfreereport-0.8.3.tar.gz
```

### 2.3.4 The Files

After unpacking the JFreeReport archive, you will see the following files and directories:

File:	Description:
<code>ChangeLog</code>	A text file containing notes on changes in each version of JFreeReport.
<code>README</code>	A text file containing important information about JFreeReport. Read this first!!!
<code>ant/</code>	A directory containing an Ant build script for JFreeReport.
<code>checkstyle/</code>	A directory containing a Checkstyle property file.
<code>jfreereport-0.8.3-demo.jar</code>	The runtime jar file for the JFreeReport demonstration applications.
<code>jfreereport-0.8.3.jar</code>	The runtime jar file for the JFreeReport class library.
<code>lib/</code>	A directory containing runtime jar files for the libraries that JFreeReport depends on.
<code>license-LGPL.txt</code>	The full text of the GNU Lesser General Public License.
<code>resource/</code>	A directory containing reference info and sample reports.
<code>source/</code>	A directory containing the JFreeReport source code.

You should take a little time to familiarise yourself with the contents of the distribution and, in particular, read the `README` file.

## 2.4 Running the Demonstration Application

Several demonstration applications are included in the JFreeReport distribution. These are intended to illustrate some of the features supported by JFreeReport.

To run the main demo, switch to the `jfreereport-0.8.3` directory and type the following command:

```
java -jar jfreereport-0.8.3-demo.jar
```

This automatically sets up the appropriate classpath, and invokes the main demo class (`com.jrefinery.report.demo.JFreeReportDemo`). An alternative way to run the same application is to specify the classpath manually using the following command (all on one line):

```
java -classpath jfreereport-0.8.3-demo.jar:jfreereport-0.8.3.jar
:lib/jcommon-0.8.2.jar:lib/itext-0.99.jar:lib/gnujarp.jar
:lib/bsh-1.2b6.jar:lib/pixie-0.8.0.jar
:lib/jakarta-poi-1.10.0-dev-20030222.jar
com.jrefinery.report.demo.JFreeReportDemo
```

Special points to note:

- if you are using Windows, you should use a semi-colon rather than a colon as the path separator, and a backward slash rather than a forward slash within paths;
- depending on your system setup, you may need to specify the full path for the `java` executable.

## 2.5 Compiling the Source

An Ant build script is included in the JFreeReport distribution, in the `ant` directory. This script has been tested using Ant version 1.5.3. For more information about Ant, refer to:

<http://jakarta.apache.org/ant>

To recompile the JFreeReport source code, you can use the following command:

```
ant compile
```

This creates a temporary `build` directory within the `jfreereport-0.8.3` directory, compiles the JFreeReport classes, then creates a new runtime jar file.

Similarly, to recompile the JFreeReport demonstration applications, you can use the following command:

```
ant compile-demo
```

This creates a temporary `build` directory within the `jfreereport-0.8.3` directory, compiles the JFreeReport demo classes, then creates a new runtime jar file.

## 2.6 Generating the Javadoc Documentation

The JFreeReport source code contains *Javadoc comments*. You can use the `javadoc` tool to generate HTML documentation files directly from the source code.<sup>4</sup>

To generate the documentation, you can use the Ant build script:

```
ant javadoc
```

This will create a new `javadoc` directory (located within the `jfreereport-0.8.3` directory) containing the Javadoc reference information for JFreeReport. To view the Javadocs, open the `javadoc/index.html` file in your favourite web browser.

---

<sup>4</sup>The Javadoc HTML files can be viewed on-line at the JFreeReport home page.

## 3 Getting Started

### 3.1 Introduction

In this section, two sample applications are presented as an introduction to using JFreeReport:

- `HelloWorld.java` – a minimal application that illustrates just the basic steps in creating and displaying a report;
- `SwingIconsDemo.java` – a more complete example that demonstrates the use of various formatting options provided by JFreeReport;

These examples should be sufficient to get you started using JFreeReport in your own applications.

### 3.2 The Basic Steps

#### 3.2.1 Overview

In standard usage, there are three major tasks in generating reports with JFreeReport:

- arrange for some data that can be accessed via the `TableModel` interface (that is, the model used by Swing's `JTable` class);
- create a `JFreeReport` object that will control the formatting of the generated report;
- link the data (`TableModel`) with the `JFreeReport` instance and pass the report to a print preview frame for presentation to the user;

Other variations are possible (for example, sending a report directly to file *without* previewing it first) but the above steps represent the most common scenario.

#### 3.2.2 Step 1: The Data

JFreeReport is designed to work with data that is accessible via the `TableModel` interface. If you are already familiar with using Swing's `JTable` class, then this will cause you no trouble. If not, I recommend that you find out more about Swing's `JTable` class before you tackle JFreeReport. Tutorials can be found at Sun's Java website:

<http://java.sun.com/>

If you want to generate reports using data accessed via JDBC, you will be pleased to know that JFreeReport includes code for generating a `TableModel` instance from a `JDBC ResultSet`.

### 3.2.3 Step 2: The Report Definition

All report formatting information is recorded by an instance of the `JFreeReport` class. You have two options for creating this instance:

- write a report definition in XML format, and get `JFreeReport` to parse the definition and create a corresponding `JFreeReport` object;
- create a `JFreeReport` instance in code, and populate the report bands and elements using the `JFreeReport` API;

Putting your report definitions into an XML file is a good idea, because it allows you to change the formatting of your reports without recompiling your application.

Creating reports in code can allow additional flexibility if you want to vary the format of your reports at run-time, but has the disadvantage that your report formatting is “hard-coded” into your application.

### 3.2.4 Step 3: Previewing the Report

The final step—previewing the report—is mostly taken care of by the `JFreeReport` library. The print preview frame will display the report on-screen, provide controls to page forward and backward through the report, zoom in and zoom out, print the report, or export to a number of different file formats including PDF, HTML, CSV, Excel and XML.

## 3.3 Sample Application: Hello World

### 3.3.1 Overview

The `HelloWorld.java` application (included in the distribution) provides a basic overview of the steps for creating and viewing reports with `JFreeReport`.

### 3.3.2 The Data

The data for this example is a simple table:

Column1:	Column2:
Hello	World!

It is created using the following code:

```
private TableModel createData()
{
    Object[] columnNames = new String[] { "Column1", "Column2" };
    DefaultTableModel result = new DefaultTableModel(columnNames, 1);
    result.setValueAt("Hello", 0, 0);
    result.setValueAt("World!", 0, 1);
    return result;
}
```

The column names are important, since they will be referenced by elements within the report definition (see the next section).

### 3.3.3 The Report

The `JFreeReport` instance used to control the format of the “Hello World” report is created in code, so that the demo is self-contained:

```
private JFreeReport createReportDefinition()
{
    JFreeReport report = new JFreeReport();
    report.setName("A Very Simple Report");

    TextElement t1 = ItemFactory.createStringElement(
        "T1",
        new Rectangle2D.Double(0.0, 0.0, 150.0, 20.0),
        Color.black,
        ElementAlignment.LEFT.getOldAlignment(),
        ElementAlignment.MIDDLE.getOldAlignment(),
        null, // font
        "-", // null string
        "Column1"
    );

    report.getItemBand().addElement(t1);

    TextElement t2 = ItemFactory.createStringElement(
        "T2",
        new Rectangle2D.Double(200.0, 0.0, 150.0, 20.0),
        Color.black,
        ElementAlignment.LEFT.getOldAlignment(),
        ElementAlignment.MIDDLE.getOldAlignment(),
        null, // font
        "-", // null string
        "Column2"
    );

    report.getItemBand().addElement(t2);
    return report;
}
```

To understand what is going on here, you need to know that the report layout is controlled by *elements* that are added to *bands* within the report. A report consists of a number of (possibly empty) bands, including:

- the *report header* – printed once at the beginning of the report;
- the *report footer* – printed once at the end of the report;
- the *page header* – printed at the top of each page;
- the *page footer* – printed at the bottom of each page;
- the *item band* – printed once for each row of data in the `TableModel`;
- an additional *group header* and/or *group footer* for each *group* defined in the report.

In the `HelloWorld.java` application, only the item band is populated—the other bands remain empty (the default). A text element is added for each column of data in the `TableModel`.

The `ItemFactory` class is used to create individual elements. Each element carries its own position, alignment, font settings and other presentation attributes. Notice also that each element in the example is tied back to the `TableModel` by a reference to a column name in the table.

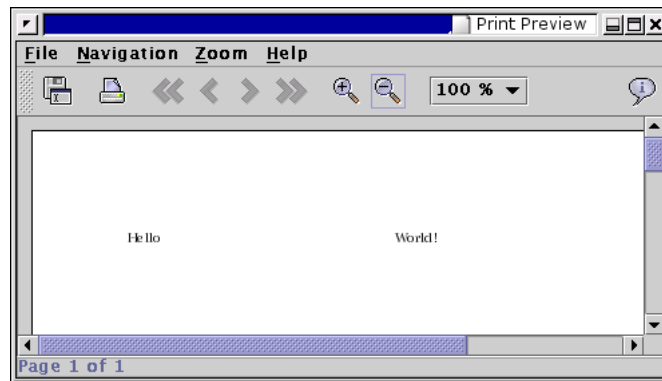
### 3.3.4 Previewing the Report

To preview this report, the data is linked with the report (using the `setData(...)` method), and the report is passed to a print preview frame:

```
TableModel data = createData();
JFreeReport report = createReportDefinition();

report.setData(data);
try
{
    PreviewFrame preview = new PreviewFrame(report);
    preview.pack();
    preview.setVisible(true);
}
catch (ReportProcessingException e)
{
    Log.error("Failed to generate report ", e);
}
```

If you run the demo, you should see the following preview frame:



Note: to run the demo from the precompiled jar files in the JFreeReport distribution, you can use the following command:

```
java -classpath jfreereport-0.8.3-demo.jar
:jfreereport-0.8.3.jar:lib/jcommon-0.8.2.jar
:lib/gnujarp.jar:lib/bsh-1.2b6.jar:lib/pixie-0.8.0.jar
:lib/itext-0.99.jar:lib/jakarta-poi-1.10.0-dev-20030222.jar
com.jrefinery.report.demo.HelloWorld
```

As usual, colons should be replaced by semi-colons if you are using Windows.

## 3.4 Sample Application: SwingIconsDemo

### 3.4.1 Overview

The `SwingIconsDemo.java` application (included in the distribution) generates a report listing the icons included in Sun's *Java Look and Feel Graphics Repository*, a collection of standard icons distributed by Sun Microsystems for use with Java. You will need to download this icon set, and agree to Sun's terms and conditions, before you will be able to run the demo application.

This example uses an XML report definition file to create the `JFreeReport` instance that controls the final report output.

### 3.4.2 Running the Demo

You may find the material in the following sections easier to follow if you have already tried out the demo application. In this section I describe how to run the `SwingIconsDemo` application.

The `SwingIconsDemo.java` source file is included in the JFreeReport distribution, along with the report template file (`swing-icons.xml`).

As mentioned previously, running this demo requires you to first download the *Java Look and Feel Graphics Repository*, a collection of standard icons (provided by Sun) for use in Swing applications. These icons are contained in a jar file (`jlfgr-1.0.jar`) that you can download<sup>5</sup> from:

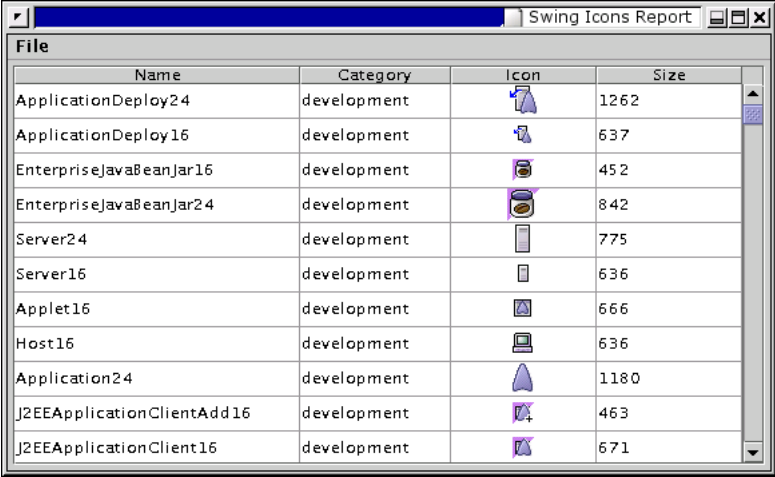
<http://developer.java.sun.com/developer/techDocs/hi/repository/>

When you run the `SwingIconsDemo` application, it will look for this file on the class path. I recommend that you place the file in the `jfreereport-0.8.3` directory, then use the following command to run the demo:












```
java -classpath jlfgr-1.0.zip:jfreereport-0.8.3-demo.jar
:jfreereport-0.8.3.jar:lib/jcommon-0.8.2.jar
:lib/gnujarp.jar:lib/bsh-1.2b6.jar:lib/pixie-0.8.0.jar
:lib/iText-0.99.jar:lib/jakarta-poi-1.10.0-dev-20030222.jar
com.jrefinery.report.demo.SwingIconsDemo
```

### 3.4.3 The Data

The `SwingIconsDemo` class uses the `SwingIconsDemoTableModel` class to store the data for the report. This table model contains code for reading the icons from the icon file (`jlfgr-1.0.jar`). The data is then displayed in a simple user interface using a `JTable`:



The screenshot shows a window titled "Swing Icons Report". Inside the window is a table with four columns: "Name", "Category", "Icon", and "Size". The table lists various Java Swing icons and their properties.

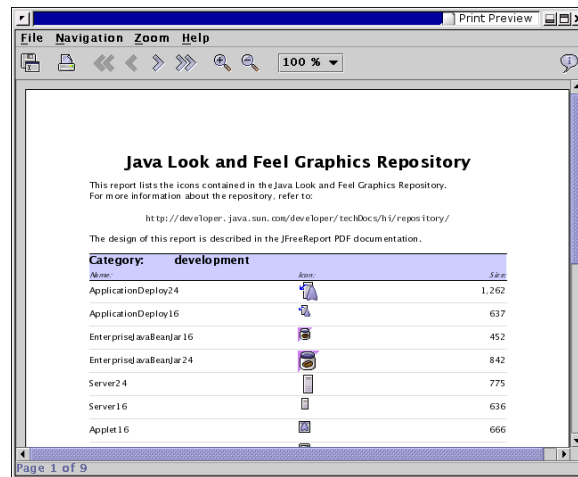
Name	Category	Icon	Size
ApplicationDeploy24	development		1262
ApplicationDeploy16	development		637
EnterpriseJavaBeanJar16	development		452
EnterpriseJavaBeanJar24	development		842
Server24	development		775
Server16	development		636
Applet16	development		666
Host16	development		636
Application24	development		1180
J2EEApplicationClientAdd16	development		463
J2EEApplicationClient16	development		671

Some important points to note about the dataset:

<sup>5</sup>Subject to acceptance of Sun's licence agreement.

- the column names are *Name*, *Category*, *Icon* and *Size*. These names are used in the XML report template file to reference data items;
- the data is sorted by the *Category* column. This is important since the data will be grouped by *Category*. JFreeReport does not currently perform any sorting, so you need to provide the data pre-sorted.

When you run the **SwingIconsDemo** application, you will see a menu item that allows you to display a print preview window. When you preview the sample report, you will see output like this:



The format of the report is controlled by an XML report template file (`swing-icons.xml`), described in the next section.

## 3.5 The Report Template File

### 3.5.1 Overview

The `swing-icons.xml` file contains a report definition in the format required by JFreeReport's "extended" parser (a new feature in version 0.8.1). This parser is more powerful and flexible than the former "simple" parser, at the expense of being more verbose. The old format is still supported for backward compatibility, and a utility is included with JFreeReport to convert reports from the old format to the new format.

Here is a skeleton version of the file (the missing pieces, marked with comments, are reproduced in later sections):

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE report PUBLIC "-//JFreeReport//DTD report definition//EN//extended"
    "http://jfreereport.sourceforge.net/extreport.dtd">

<report-definition name="First Report">

  <!-- * INSERT PARSER CONFIGURATION HERE * -->

  <!-- * INSERT REPORT CONFIGURATION HERE * -->

  <!-- * INSERT STYLES HERE * -->

  <!-- * INSERT TEMPLATES HERE * -->

  <!-- * INSERT FUNCTIONS HERE * -->

  <!-- ***** -->

```

```

<!-- * REPORT DESCRIPTION * -->
<!-- ***** -->

<report-description>

    <!-- INSERT REPORT HEADER HERE -->

    <!-- INSERT REPORT FOOTER HERE -->

    <!-- INSERT PAGE HEADER HERE -->

    <!-- INSERT PAGE FOOTER HERE -->

    <!-- INSERT GROUP DEFINITIONS HERE -->

    <!-- INSERT ITEM BAND HERE -->

</report-description>

</report-definition>

```

The report name is required, and is made available within the report as a *report property* (`report.name`).

### 3.5.2 The Parser Configuration

The *parser configuration* provides a mechanism for customising the behaviour of the report definition parser:

```

<!-- ***** -->
<!-- * PARSER CONFIGURATION * -->
<!-- ***** -->
<parser-config>
  <object-factory class="com.jrefinery.report.io.ext.factory.datasource.DataSourceCollector"/>
  <object-factory class="com.jrefinery.report.io.ext.factory.objects.DefaultClassFactory"/>
  <element-factory class="com.jrefinery.report.io.ext.factory.elements.DefaultElementFactory"/>
  <stylekey-factory class="com.jrefinery.report.io.ext.factory.stylekey.DefaultStyleKeyFactory"/>
  <stylekey-factory class="com.jrefinery.report.io.ext.factory.stylekey.PageableLayoutStyleKeyFactory"/>
  <template-factory class="com.jrefinery.report.io.ext.factory.templates.DefaultTemplateCollection"/>
  <datasource-factory class="com.jrefinery.report.io.ext.factory.datasource.DefaultDataSourceFactory"/>
</parser-config>

```

Most of the time, you will just copy and paste this default configuration.

### 3.5.3 The Report Configuration

The *report configuration* allows you to control certain aspects of the way that JFreeReport works. In the example, the default page format and the initial height and width of the preview frame is specified:

```

<!-- ***** -->
<!-- * REPORT CONFIGURATION * -->
<!-- ***** -->
<report-config>
  <defaultpageformat orientation="portrait"
    pageformat="LETTER"
    topmargin="72"
    bottommargin="72"
    leftmargin="72"
    rightmargin="72"/>

  <configuration>
    <property name="com.jrefinery.report.preview.PreferredHeight">480</property>
    <property name="com.jrefinery.report.preview.PreferredWidth">640</property>
  </configuration>
</report-config>

```

All the valid `pageformat` options are specified in the `PageFormatFactory` class. The margins are specified in points (1/72 inch).

Additional details on the available configuration parameters can be found in the documentation for the [ReportConfiguration](#) class.

### 3.5.4 Styles

The *Styles* element is empty in this example:

```
<!-- ***** -->
<!-- * STYLES * -->
<!-- ***** -->
<styles>
</styles>
```

### 3.5.5 Templates

The *Templates* element is empty in this example:

```
<!-- ***** -->
<!-- * TEMPLATES * -->
<!-- ***** -->
<templates>
</templates>
```

### 3.5.6 Report Functions

The example uses several *report functions* to calculate summary information for display on the report. It is necessary to declare functions within the XML report template file—individual report elements can then reference particular functions by name.

When you declare a function, you give it a name and specify the Java class that implements the function. Most functions also require some properties to be specified (you should refer to the documentation or source code for the class that implements the function to find out which properties are required).

Here is the XML used to define the functions used in the sample report:

```
<!-- ***** -->
<!-- * FUNCTIONS * -->
<!-- ***** -->
<functions>

  <property-ref name="report.date"/>

  <function name="PageNumber" class="com.jrefinery.report.function.PageFunction">
  </function>

  <function name="GroupCount" class="com.jrefinery.report.function.ItemCountFunction">
    <properties>
      <property name="field">Name</property>
      <property name="group">Category</property>
    </properties>
  </function>

  <function name="GroupSum" class="com.jrefinery.report.function.ItemSumFunction">
    <properties>
      <property name="field">Size</property>
      <property name="group">Category</property>
    </properties>
  </function>

  <function name="OverallCount" class="com.jrefinery.report.function.ItemCountFunction">
    <properties>
      <property name="field">Name</property>
    </properties>
  </function>

  <function name="OverallSum" class="com.jrefinery.report.function.ItemSumFunction">
    <properties>
      <property name="field">Size</property>
    </properties>
  </function>
</functions>
```

The first entry in the function declarations marks a report property for access. The (**report.date**) property is automatically set up by JFreeReport at report generation time, and it returns the current date. In the example, the report date is displayed in the page header.

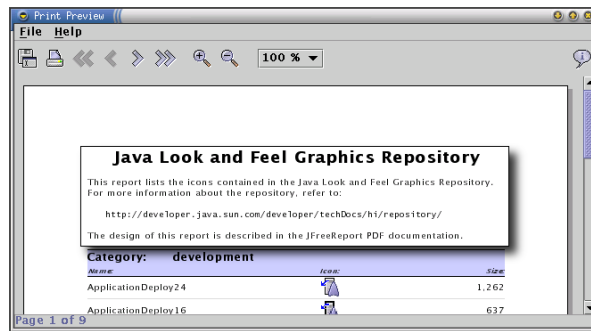
The first function (**PageNumber**) returns the current page number. It does not require any properties to be set. In the example report, the page number is displayed in the page footer.

There are two functions declared to count report items. The first (`GroupCount`) counts items within the `Category` group, and is used in the group footer. The second (`OverallCount`) counts items for the entire report (simply by omitting the `group` property), and is used in the report footer.

There are two functions declared to sum the icon size field. The first (`GroupSum`) calculates a running total within the `Category` group, and is used in the group footer. The second (`OverallSum`) calculates a running total for the entire report, and is used in the report footer.

### 3.5.7 The Report Header

The *report header* is a band that is printed once at the beginning of a report. In our example, the report header contains five labels (fixed text items), highlighted in the figure below:



The minimum height of the band is 112 points. The default font is plain `SansSerif`, at 10 points. Here is the XML used to define the report header:

```
<!-- ===== -->
<!-- = REPORT HEADER = -->
<!-- ===== -->
<report-header name="report-header-band">

  <style>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">112.0</basic-object>
      <basic-object name="width" class="java.lang.Double">0.0</basic-object>
    </compound-key>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-size">10</basic-key>
    <basic-key name="font-bold">>false</basic-key>
    <basic-key name="font-italic">>false</basic-key>
    <basic-key name="pagebreak-after">>false</basic-key>
  </style>

  <default-style>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-size">10</basic-key>
    <basic-key name="font-bold">>false</basic-key>
    <basic-key name="font-italic">>false</basic-key>
  </default-style>

  <element name="title1" type="text/plain">
    <style>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="y">2.0</basic-object>
      </compound-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">18.0</basic-object>
        <basic-object name="width" class="java.lang.Double">100.0</basic-object>
      </compound-key>
      <basic-key name="dynamic_height">>false</basic-key>
      <basic-key name="alignment">center</basic-key>
      <basic-key name="valignment">bottom</basic-key>
      <basic-key name="font-size">18</basic-key>
      <basic-key name="font-bold">>true</basic-key>
      <basic-key name="font-italic">>false</basic-key>
      <basic-key name="paint">black</basic-key>
    </style>
  </element>
</report-header>
```

```

<template references="label">
  <basic-object name="content">Java Look and Feel Graphics Repository</basic-object>
  <basic-object name="nullValue">null</basic-object>
</template>
</element>

<element name="description1" type="text/plain">
  <style>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">0.0</basic-object>
      <basic-object name="y">32.0</basic-object>
    </compound-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">10.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
    </compound-key>
    <basic-key name="dynamic_height">false</basic-key>
    <basic-key name="alignment">left</basic-key>
    <basic-key name="valignment">bottom</basic-key>
    <basic-key name="paint">black</basic-key>
  </style>
  <template references="label">
    <basic-object name="content">This report lists the icons contained in the Java Look and Feel Graphics Repository.</basic-object>
    <basic-object name="nullValue">null</basic-object>
  </template>
</element>

<element name="description2" type="text/plain">
  <style>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">0.0</basic-object>
      <basic-object name="y">44.0</basic-object>
    </compound-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">10.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
    </compound-key>
    <basic-key name="dynamic_height">false</basic-key>
    <basic-key name="alignment">left</basic-key>
    <basic-key name="valignment">bottom</basic-key>
    <basic-key name="paint">black</basic-key>
  </style>
  <template references="label">
    <basic-object name="content">For more information about the repository, refer to:</basic-object>
    <basic-object name="nullValue">null</basic-object>
  </template>
</element>

<element name="url1" type="text/plain">
  <style>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">0.0</basic-object>
      <basic-object name="y">68.0</basic-object>
    </compound-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">10.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
    </compound-key>
    <basic-key name="dynamic_height">false</basic-key>
    <basic-key name="alignment">center</basic-key>
    <basic-key name="valignment">bottom</basic-key>
    <basic-key name="font">Monospaced</basic-key>
    <basic-key name="font-size">9</basic-key>
    <basic-key name="paint">black</basic-key>
  </style>
  <template references="label">
    <basic-object name="content">http://developer.java.sun.com/developer/techDocs/hi/repository/</basic-object>
    <basic-object name="nullValue">null</basic-object>
  </template>
</element>

<element name="description3" type="text/plain">
  <style>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">0.0</basic-object>
      <basic-object name="y">92.0</basic-object>
    </compound-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">10.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
    </compound-key>
    <basic-key name="dynamic_height">false</basic-key>
    <basic-key name="alignment">left</basic-key>
    <basic-key name="valignment">bottom</basic-key>
    <basic-key name="paint">black</basic-key>
  </style>
  <template references="label">
    <basic-object name="content">The design of this report is described in the JFreeReport PDF documentation.</basic-object>
    <basic-object name="nullValue">null</basic-object>
  </template>
</element>
</report-header>

```

Special points to note:

- the x and y coordinates for each element are measured in points (1/72

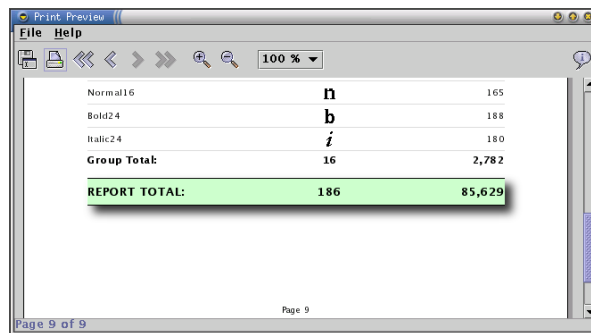
inch) relative to the band's origin;

- the y-values increase as you move down the band / page;
- the width of each label is 100 percent (percentages are coded as negative values), which means that the element area extends across the full width of the report. This is particularly important for the first label, which is centered within its area, and so appears at the center of the band;
- the default font is overridden for first and fourth labels. For the other labels, the default font specified for the header is used.

Although the example displays only labels, you can use any report element in a report header, including those that display function values.

### 3.5.8 The Report Footer

The *report footer* is a band that is printed once at the end of a report. In the example, the report footer contains a label and two number fields that get their values from specific report functions. It also contains two lines and a rectangle, use to enhance the appearance of the report. The footer is highlighted in the figure below:



Here is the XML used to define the report footer:

```
<!-- ===== -->
<!-- = REPORT FOOTER = -->
<!-- ===== -->
<report-footer name="report-footer-band">

  <style>
    <basic-key name="font-bold">true</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">30.0</basic-object>
      <basic-object name="width" class="java.lang.Double">0.0</basic-object>
    </compound-key>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-size">12</basic-key>
    <basic-key name="font-italic">false</basic-key>
    <basic-key name="pagebreak-before">false</basic-key>
  </style>

  <default-style>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-size">12</basic-key>
    <basic-key name="font-bold">true</basic-key>
    <basic-key name="font-italic">false</basic-key>
  </default-style>

  <element name="rect1" type="shape/generic">
    <style>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="y">0.0</basic-object>
      </compound-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
```

```

        <basic-object name="height" class="java.lang.Double">-100.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
    </compound-key>
    <basic-key name="stroke" class="java.awt.BasicStroke">0.0</basic-key>
    <basic-key name="paint">#ccffcc</basic-key>
    <basic-key name="draw-shape">>false</basic-key>
    <basic-key name="fill-shape">>true</basic-key>
    <basic-key name="scale">>true</basic-key>
    <basic-key name="keepAspectRatio">>false</basic-key>
</style>
<datasource type="StaticDataSource">
    <compound-object name="value" class="java.awt.geom.Rectangle2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="height">100.0</basic-object>
        <basic-object name="width">100.0</basic-object>
        <basic-object name="y">0.0</basic-object>
    </compound-object>
</datasource>
</element>

<element name="line1" type="shape/generic">
    <style>
        <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
            <basic-object name="x">0.0</basic-object>
            <basic-object name="y">0.0</basic-object>
        </compound-key>
        <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
            <basic-object name="height" class="java.lang.Double">0.0</basic-object>
            <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
        </compound-key>
        <basic-key name="paint">black</basic-key>
        <basic-key name="stroke" class="java.awt.BasicStroke">0.5</basic-key>
        <basic-key name="draw-shape">>true</basic-key>
        <basic-key name="fill-shape">>false</basic-key>
        <basic-key name="keepAspectRatio">>false</basic-key>
        <basic-key name="scale">>true</basic-key>
    </style>
    <datasource type="StaticDataSource">
        <compound-object name="value" class="java.awt.geom.Line2D$Float">
            <basic-object name="x2">100.0</basic-object>
            <basic-object name="x1">0.0</basic-object>
            <basic-object name="y2">0.0</basic-object>
            <basic-object name="y1">0.0</basic-object>
        </compound-object>
    </datasource>
</element>

<element name="line2" type="shape/generic">
    <style>
        <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
            <basic-object name="x">0.0</basic-object>
            <basic-object name="y">30.0</basic-object>
        </compound-key>
        <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
            <basic-object name="height" class="java.lang.Double">0.0</basic-object>
            <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
        </compound-key>
        <basic-key name="stroke" class="java.awt.BasicStroke">0.5</basic-key>
        <basic-key name="paint">black</basic-key>
        <basic-key name="draw-shape">>true</basic-key>
        <basic-key name="fill-shape">>false</basic-key>
        <basic-key name="scale">>true</basic-key>
        <basic-key name="keepAspectRatio">>false</basic-key>
    </style>
    <datasource type="StaticDataSource">
        <compound-object name="value" class="java.awt.geom.Line2D$Float">
            <basic-object name="x2">100.0</basic-object>
            <basic-object name="x1">0.0</basic-object>
            <basic-object name="y2">30.0</basic-object>
            <basic-object name="y1">30.0</basic-object>
        </compound-object>
    </datasource>
</element>

<element name="reportTotalLabel" type="text/plain">
    <style>
        <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
            <basic-object name="x">0.0</basic-object>
            <basic-object name="y">10.0</basic-object>
        </compound-key>
        <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
            <basic-object name="height" class="java.lang.Double">12.0</basic-object>
            <basic-object name="width" class="java.lang.Double">-50.0</basic-object>
        </compound-key>
        <basic-key name="alignment">left</basic-key>
        <basic-key name="valignment">bottom</basic-key>
        <basic-key name="dynamic_height">>false</basic-key>
        <basic-key name="paint">black</basic-key>
    </style>
    <template references="label">
        <basic-object name="content">REPORT TOTAL:</basic-object>
        <basic-object name="nullValue">null</basic-object>
    </template>
</element>

<element name="reportCountField" type="text/plain">
    <style>
        <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
            <basic-object name="x">-50.0</basic-object>

```

```

        <basic-object name="y">10.0</basic-object>
      </compound-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">9.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-30.0</basic-object>
      </compound-key>
      <basic-key name="paint">black</basic-key>
      <basic-key name="alignment">center</basic-key>
      <basic-key name="valignment">bottom</basic-key>
      <basic-key name="dynamic_height">false</basic-key>
    </style>
    <template references="number-field">
      <basic-object name="field">OverallCount</basic-object>
      <basic-object name="format">#0</basic-object>
      <basic-object name="nullValue">-</basic-object>
    </template>
  </element>

  <element name="reportSumField" type="text/plain">
    <style>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">-80.0</basic-object>
        <basic-object name="y">10.0</basic-object>
      </compound-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">9.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-20.0</basic-object>
      </compound-key>
      <basic-key name="paint">black</basic-key>
      <basic-key name="alignment">right</basic-key>
      <basic-key name="valignment">bottom</basic-key>
      <basic-key name="dynamic_height">false</basic-key>
    </style>
    <template references="number-field">
      <basic-object name="field">OverallSum</basic-object>
      <basic-object name="format">#,##0</basic-object>
      <basic-object name="nullValue">-</basic-object>
    </template>
  </element>
</report-footer>

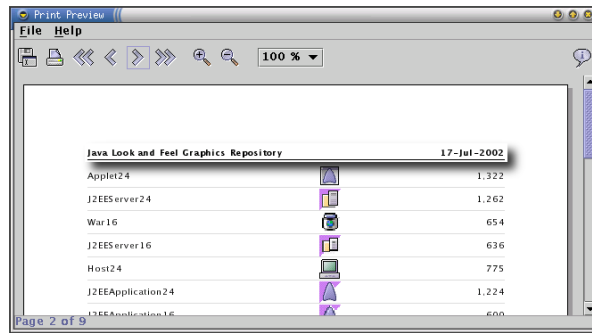
```

Special points to note:

- the lines and shapes are declared first, so that they are drawn first (to avoid obscuring the other report elements in the band);
- the coordinates for the `line` elements both represent a single point, in this example. This is a special case, where JFreeReport draws a horizontal line passing through the point and covering the full width of the band;
- the `number-field` elements specify the name of the function that supplies the element value. Each function must be declared elsewhere in the report definition (see section 3.5.6);
- for the `number-field` report elements, you can define a format string to format the output. Internally, this is passed to an instance of Java's `NumberFormat` class to control the formatting.

### 3.5.9 The Page Header

The *page header* appears at the top of every page, with the possible exception of the first and last pages. In the example, the page header contains a label and a date field that displays the current date. This is highlighted in the figure below:



Here is the XML used to define the page header:

```
<!-- ===== -->
<!-- = PAGE HEADER = -->
<!-- ===== -->
<page-header name="anonymousBand@1319318">

<style>
  <basic-key name="font-bold">true</basic-key>
  <basic-key name="display-on-lastpage">true</basic-key>
  <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
    <basic-object name="height" class="java.lang.Double">18.0</basic-object>
    <basic-object name="width" class="java.lang.Double">0.0</basic-object>
  </compound-key>
  <basic-key name="font-size">9</basic-key>
  <basic-key name="display-on-firstpage">>false</basic-key>
  <basic-key name="font">SansSerif</basic-key>
  <basic-key name="font-italic">>false</basic-key>
</style>

<default-style>
  <basic-key name="font-bold">true</basic-key>
  <basic-key name="font-size">9</basic-key>
  <basic-key name="font">SansSerif</basic-key>
  <basic-key name="font-italic">>false</basic-key>
</default-style>

<element name="@anonymous" type="shape/generic">
  <style>
    <basic-key name="stroke" class="java.awt.BasicStroke">0.0</basic-key>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">0.0</basic-object>
      <basic-object name="y">0.0</basic-object>
    </compound-key>
    <basic-key name="draw-shape">>false</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">-100.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
    </compound-key>
    <basic-key name="keepAspectRatio">>false</basic-key>
    <basic-key name="scale">true</basic-key>
    <basic-key name="paint">#afafaf</basic-key>
    <basic-key name="fill-shape">true</basic-key>
  </style>
  <datasource type="StaticDataSource">
    <compound-object name="value" class="java.awt.geom.Rectangle2D$Float">
      <basic-object name="x">0.0</basic-object>
      <basic-object name="height">100.0</basic-object>
      <basic-object name="width">100.0</basic-object>
      <basic-object name="y">0.0</basic-object>
    </compound-object>
  </datasource>
</element>

<element name="@anonymous" type="text/plain">
  <style>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">0.0</basic-object>
      <basic-object name="y">0.0</basic-object>
    </compound-key>
    <basic-key name="dynamic_height">>false</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">14.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-50.0</basic-object>
    </compound-key>
    <basic-key name="paint">black</basic-key>
    <basic-key name="alignment">left</basic-key>
    <basic-key name="valignment">bottom</basic-key>
  </style>
  <template references="label">
    <basic-object name="nullValue">>null</basic-object>
    <basic-object name="content">Java Look and Feel Graphics Repository</basic-object>
  </template>
</element>
```

```

<element name="@anonymous" type="text/plain">
  <style>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">-80.0</basic-object>
      <basic-object name="y">0.0</basic-object>
    </compound-key>
    <basic-key name="dynamic_height">false</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">14.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-20.0</basic-object>
    </compound-key>
    <basic-key name="paint">black</basic-key>
    <basic-key name="alignment">right</basic-key>
    <basic-key name="valignment">bottom</basic-key>
  </style>
  <template references="date-field">
    <basic-object name="nullValue">-</basic-object>
    <basic-object name="format">d-MMM-yyyy</basic-object>
    <basic-object name="field">report.date</basic-object>
  </template>
</element>

<element name="@anonymousf" type="shape/generic">
  <style>
    <basic-key name="stroke" class="java.awt.BasicStroke">2.0</basic-key>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">0.0</basic-object>
      <basic-object name="y">16.0</basic-object>
    </compound-key>
    <basic-key name="draw-shape">true</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">0.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
    </compound-key>
    <basic-key name="keepAspectRatio">false</basic-key>
    <basic-key name="scale">true</basic-key>
    <basic-key name="paint">#cfcfcf</basic-key>
    <basic-key name="fill-shape">false</basic-key>
  </style>
  <datasource type="StaticDataSource">
    <compound-object name="value" class="java.awt.geom.Line2D$Float">
      <basic-object name="x2">100.0</basic-object>
      <basic-object name="x1">0.0</basic-object>
      <basic-object name="y2">16.0</basic-object>
      <basic-object name="y1">16.0</basic-object>
    </compound-object>
  </datasource>
</element>
</page-header>

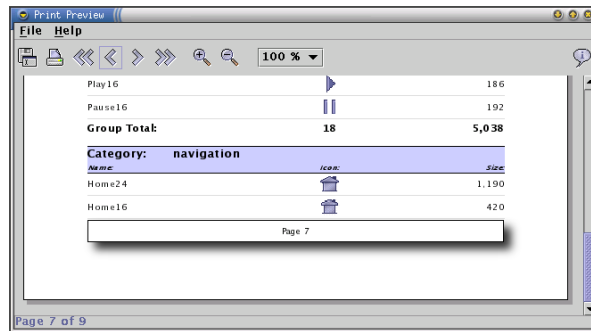
```

Special points to note:

- the page header is suppressed on the first page, since it would conflict with the report header;
- the `date-field` element references (by name) the report date property. This property is declared elsewhere in the report definition (see section 3.5.6);
- the `date-field` element has a format attribute to control date formatting. This is passed to an instance of Java's `SimpleDateFormat` class to control the formatting.

### 3.5.10 The Page Footer

The *page footer* appears at the bottom of every page, with the possible exception of the first and last pages. In the example, the page footer contains just a number field that displays the page number. This is highlighted in the figure below:



Here is the XML used to define the page footer:

```
<!-- ===== -->
<!-- = PAGE FOOTER = -->
<!-- ===== -->
<page-footer name="anonymousBand@7205769">

  <style>
    <basic-key name="font-bold">false</basic-key>
    <basic-key name="display-on-lastpage">true</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">14.0</basic-object>
      <basic-object name="width" class="java.lang.Double">0.0</basic-object>
    </compound-key>
    <basic-key name="font-size">9</basic-key>
    <basic-key name="display-on-firstpage">true</basic-key>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-italic">false</basic-key>
  </style>

  <default-style>
    <basic-key name="font-bold">false</basic-key>
    <basic-key name="font-size">9</basic-key>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-italic">false</basic-key>
  </default-style>

  <element name="@anonymous10" type="text/plain">
    <style>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="y">3.0</basic-object>
      </compound-key>
      <basic-key name="dynamic_height">false</basic-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">9.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
      </compound-key>
      <basic-key name="paint">black</basic-key>
      <basic-key name="alignment">center</basic-key>
      <basic-key name="valignment">bottom</basic-key>
    </style>
    <template references="number-field">
      <basic-object name="nullValue">null</basic-object>
      <basic-object name="format">Page #0</basic-object>
      <basic-object name="field">PageNumber</basic-object>
    </template>
  </element>
</page-footer>
```

Special points to note:

- the page footer can be suppressed on the first or last pages, to avoid conflict with the report header and/or the report footer. In this example, the page footer is displayed on all pages;
- the elements using **number-field** templates reference (by name) the function that returns the page number. This function is declared elsewhere in the report definition (see section 3.5.6);
- the **number-field** element has a **format** attribute to control numberformatting. This is passed to an instance of Java's **NumberFormat** class to control formatting.

### 3.5.11 Report Groups

Report groups are used to aggregate report items according to the values in certain fields. In this example, only one group is defined (on the **Category** field) in addition to the default group, but it is possible to create multiple (nested) groups with JFreeReport.

For each group, you can define a *group header* that is printed at the start of each group instance. Likewise, you can define a *group footer* that is printed at the end of each group instance.

The XML used to define the **Category** group is:

```
<!-- ===== -->
<!-- = GROUPS = -->
<!-- ===== -->
<groups>

  <!-- default group -->
  <group name="default">
    <fields>
    </fields>
    <group-header name="anonymousBand@4732779">
      <style>
      </style>
    </group-header>
    <group-footer name="anonymousBand@6100951">
      <style>
      </style>
    </group-footer>
  </group>

  <!-- category group -->
  <group name="Category">
    <fields>
    <field>Category</field>
    </fields>

    <!-- INSERT GROUP HEADER HERE -->

    <!-- INSERT GROUP FOOTER HERE -->

  </group>
</groups>
```

An important point to note is that the data in the **TableModel** should be *pre-sorted* according to the fields defined in the report groups, otherwise you will get some unusual results. JFreeReport does not perform any sorting itself.

As mentioned already, JFreeReport will print a *group header* every time a new group starts, and a *group footer* every time a group ends. The header and footer definitions should be inserted at the points indicated in the XML fragment above.

The group header is highlighted in the figure below:

The screenshot shows a 'Print Preview' window of a JFreeReport. The table displayed has the following data:

Play16		186
Pause16		192
<b>Group Total:</b>	<b>18</b>	<b>5,038</b>
<b>Category: navigation</b>		
Home24		1,190
Home16		420

The 'Category: navigation' row is highlighted in blue. The window also shows a toolbar with navigation icons, a zoom level of 100%, and a status bar indicating 'Page 7 of 9'.

Here is the XML used to define the group header:

```
<group-header name="anonymousBand@3921842">
  <style>
    <basic-key name="font-bold">true</basic-key>
    <basic-key name="repeat-header">false</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">30.0</basic-object>
      <basic-object name="width" class="java.lang.Double">0.0</basic-object>
    </compound-key>
    <basic-key name="font-size">12</basic-key>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-italic">false</basic-key>
    <basic-key name="pagebreak-before">false</basic-key>
  </style>

  <default-style>
    <basic-key name="font-bold">true</basic-key>
    <basic-key name="font-size">12</basic-key>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-italic">false</basic-key>
  </default-style>

  <element name="@anonymous11" type="shape/generic">
    <style>
      <basic-key name="stroke" class="java.awt.BasicStroke">0.0</basic-key>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="y">0.0</basic-object>
      </compound-key>
      <basic-key name="draw-shape">false</basic-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">-100.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
      </compound-key>
      <basic-key name="keepAspectRatio">false</basic-key>
      <basic-key name="scale">true</basic-key>
      <basic-key name="paint">#ccccff</basic-key>
      <basic-key name="fill-shape">true</basic-key>
    </style>

    <datasource type="StaticDataSource">
      <compound-object name="value" class="java.awt.geom.Rectangle2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="height">100.0</basic-object>
        <basic-object name="width">100.0</basic-object>
        <basic-object name="y">0.0</basic-object>
      </compound-object>
    </datasource>
  </element>

  <element name="@anonymous12" type="shape/generic">
    <style>
      <basic-key name="stroke" class="java.awt.BasicStroke">0.5</basic-key>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="y">0.0</basic-object>
      </compound-key>
      <basic-key name="draw-shape">true</basic-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">0.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
      </compound-key>
      <basic-key name="keepAspectRatio">false</basic-key>
      <basic-key name="scale">true</basic-key>
      <basic-key name="paint">black</basic-key>
      <basic-key name="fill-shape">false</basic-key>
    </style>

    <datasource type="StaticDataSource">
      <compound-object name="value" class="java.awt.geom.Line2D$Float">
        <basic-object name="x2">100.0</basic-object>
        <basic-object name="x1">0.0</basic-object>
        <basic-object name="y2">0.0</basic-object>
        <basic-object name="y1">0.0</basic-object>
      </compound-object>
    </datasource>
  </element>

  <element name="@anonymous13" type="shape/generic">
    <style>
      <basic-key name="stroke" class="java.awt.BasicStroke">0.5</basic-key>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="y">30.0</basic-object>
      </compound-key>
      <basic-key name="draw-shape">true</basic-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">0.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
      </compound-key>
      <basic-key name="keepAspectRatio">false</basic-key>
      <basic-key name="scale">true</basic-key>
      <basic-key name="paint">black</basic-key>
      <basic-key name="fill-shape">false</basic-key>
    </style>

    <datasource type="StaticDataSource">
      <compound-object name="value" class="java.awt.geom.Line2D$Float">
        <basic-object name="x2">100.0</basic-object>
```

```

        <basic-object name="x1">0.0</basic-object>
        <basic-object name="y2">30.0</basic-object>
        <basic-object name="y1">30.0</basic-object>
    </compound-object>
</datasource>
</element>

<element name="@anonymous14" type="text/plain">
    <style>
        <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
            <basic-object name="x">0.0</basic-object>
            <basic-object name="y">3.0</basic-object>
        </compound-key>
        <basic-key name="dynamic_height">false</basic-key>
        <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
            <basic-object name="height" class="java.lang.Double">12.0</basic-object>
            <basic-object name="width" class="java.lang.Double">92.0</basic-object>
        </compound-key>
        <basic-key name="paint">black</basic-key>
        <basic-key name="alignment">left</basic-key>
        <basic-key name="valignment">bottom</basic-key>
    </style>
    <template references="label">
        <basic-object name="nullValue">null</basic-object>
        <basic-object name="content">Category:</basic-object>
    </template>
</element>

<element name="@anonymous15" type="text/plain">
    <style>
        <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
            <basic-object name="x">96.0</basic-object>
            <basic-object name="y">3.0</basic-object>
        </compound-key>
        <basic-key name="dynamic_height">false</basic-key>
        <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
            <basic-object name="height" class="java.lang.Double">12.0</basic-object>
            <basic-object name="width" class="java.lang.Double">120.0</basic-object>
        </compound-key>
        <basic-key name="paint">black</basic-key>
        <basic-key name="alignment">left</basic-key>
        <basic-key name="valignment">bottom</basic-key>
    </style>
    <template references="string-field">
        <basic-object name="nullValue"></basic-object>
        <basic-object name="field">Category</basic-object>
    </template>
</element>

<element name="@anonymous16" type="text/plain">
    <style>
        <basic-key name="dynamic_height">false</basic-key>
        <basic-key name="font-size">8</basic-key>
        <basic-key name="font-bold">false</basic-key>
        <basic-key name="font-italic">true</basic-key>
        <basic-key name="paint">black</basic-key>
        <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
            <basic-object name="x">0.0</basic-object>
            <basic-object name="y">20.0</basic-object>
        </compound-key>
        <basic-key name="alignment">left</basic-key>
        <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
            <basic-object name="height" class="java.lang.Double">9.0</basic-object>
            <basic-object name="width" class="java.lang.Double">-50.0</basic-object>
        </compound-key>
        <basic-key name="valignment">bottom</basic-key>
    </style>
    <template references="label">
        <basic-object name="nullValue">null</basic-object>
        <basic-object name="content">Name:</basic-object>
    </template>
</element>

<element name="@anonymous17" type="text/plain">
    <style>
        <basic-key name="dynamic_height">false</basic-key>
        <basic-key name="font-size">8</basic-key>
        <basic-key name="font-bold">false</basic-key>
        <basic-key name="font-italic">true</basic-key>
        <basic-key name="paint">black</basic-key>
        <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
            <basic-object name="x">-50.0</basic-object>
            <basic-object name="y">20.0</basic-object>
        </compound-key>
        <basic-key name="alignment">left</basic-key>
        <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
            <basic-object name="height" class="java.lang.Double">9.0</basic-object>
            <basic-object name="width" class="java.lang.Double">-30.0</basic-object>
        </compound-key>
        <basic-key name="valignment">bottom</basic-key>
    </style>
    <template references="label">
        <basic-object name="nullValue">null</basic-object>
        <basic-object name="content">Icon:</basic-object>
    </template>
</element>

<element name="@anonymous18" type="text/plain">
    <style>

```

```

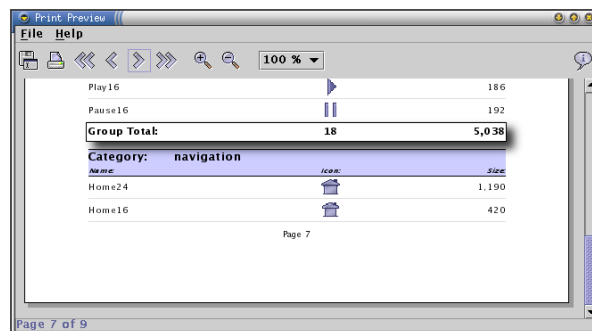
<basic-key name="dynamic_height">false</basic-key>
<basic-key name="font-size">8</basic-key>
<basic-key name="font-bold">false</basic-key>
<basic-key name="font-italic">true</basic-key>
<basic-key name="paint">black</basic-key>
<compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
  <basic-object name="x">-80.0</basic-object>
  <basic-object name="y">20.0</basic-object>
</compound-key>
<basic-key name="alignment">right</basic-key>
<compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
  <basic-object name="height" class="java.lang.Double">9.0</basic-object>
  <basic-object name="width" class="java.lang.Double">-20.0</basic-object>
</compound-key>
<basic-key name="valignment">bottom</basic-key>
</style>
<template references="label">
  <basic-object name="nullValue">null</basic-object>
  <basic-object name="content">Size</basic-object>
</template>
</element>
</group-header>

```

Some points to note:

- the `pagebreak` attribute is set to `false`. If you change this to `true`, the reporting engine will start a new page before printing the group header;
- a `string-field` template is used to display the value in the `Category` column of the report's `TableModel`;

The group footer is highlighted in the figure below:



Here is the XML used to define the group footer:

```

<group-footer name="anonymousBand08361137">
  <style>
    <basic-key name="font-bold">true</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">30.0</basic-object>
      <basic-object name="width" class="java.lang.Double">0.0</basic-object>
    </compound-key>
    <basic-key name="font-size">11</basic-key>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-italic">false</basic-key>
    <basic-key name="pagebreak-before">false</basic-key>
  </style>

  <default-style>
    <basic-key name="font-bold">true</basic-key>
    <basic-key name="font-size">11</basic-key>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-italic">false</basic-key>
  </default-style>

  <element name="@anonymous19" type="text/plain">
    <style>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="y">5.0</basic-object>
      </compound-key>
      <basic-key name="dynamic_height">false</basic-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">9.0</basic-object>

```

```

        <basic-object name="width" class="java.lang.Double">100.0</basic-object>
      </compound-key>
      <basic-key name="paint">black</basic-key>
      <basic-key name="alignment">left</basic-key>
      <basic-key name="valignment">bottom</basic-key>
    </style>
    <template references="label">
      <basic-object name="nullValue">null</basic-object>
      <basic-object name="content">Group Total:</basic-object>
    </template>
  </element>

  <element name="@anonymous1a" type="text/plain">
    <style>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">-50.0</basic-object>
        <basic-object name="y">5.0</basic-object>
      </compound-key>
      <basic-key name="dynamic_height">false</basic-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">9.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-30.0</basic-object>
      </compound-key>
      <basic-key name="paint">black</basic-key>
      <basic-key name="alignment">center</basic-key>
      <basic-key name="valignment">bottom</basic-key>
    </style>

    <template references="number-field">
      <basic-object name="nullValue"></basic-object>
      <basic-object name="format">#0</basic-object>
      <basic-object name="field">GroupCount</basic-object>
    </template>
  </element>

  <element name="@anonymous1b" type="text/plain">
    <style>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">-80.0</basic-object>
        <basic-object name="y">5.0</basic-object>
      </compound-key>
      <basic-key name="dynamic_height">false</basic-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">9.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-20.0</basic-object>
      </compound-key>
      <basic-key name="paint">black</basic-key>
      <basic-key name="alignment">right</basic-key>
      <basic-key name="valignment">bottom</basic-key>
    </style>
    <template references="number-field">
      <basic-object name="nullValue"></basic-object>
      <basic-object name="format">#.#0</basic-object>
      <basic-object name="field">GroupSum</basic-object>
    </template>
  </element>
</group-footer>

```

Some points to note:

- the two elements that use **number-field** templates reference (by name) functions declared elsewhere in the report definition (see section 3.5.6);

### 3.5.12 The Item Band

The item band is displayed once per row of data in the report's `TableModel`. In the example, the item band contains four elements:

- a string field displaying the name of the icon;
- an image field displaying the icon itself;
- a number field displaying the size of the icon image (in bytes);
- a shape element that displays a line across the page.

One instance of the item band is highlighted in the figure below:

ApplicationDeploy16		637
EnterpriseJavaBean16		452
EnterpriseJavaBean24		842
Server24		775
Server16		636
Applet16		666
Host16		636
Application24		1.180
J2EEApplicationClientAdd16		463
J2EEApplicationClient16		671

Here is the XML used to define the item band:

```
<!-- ===== -->
<!-- = ITEM BAND = -->
<!-- ===== -->
<itemband name="anonymousBand0581472">

  <!-- the band style -->
  <style>
    <basic-key name="font-bold">false</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">26.0</basic-object>
      <basic-object name="width" class="java.lang.Double">0.0</basic-object>
    </compound-key>
    <basic-key name="font-size">10</basic-key>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-italic">false</basic-key>
  </style>

  <!-- the default element style -->
  <default-style>
    <basic-key name="font">SansSerif</basic-key>
    <basic-key name="font-size">10</basic-key>
    <basic-key name="font-bold">false</basic-key>
    <basic-key name="font-italic">false</basic-key>
  </default-style>

  <element name="@anonymous1c" type="shape/generic">
    <style>
      <basic-key name="stroke" class="java.awt.BasicStroke">0.1</basic-key>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="y">25.0</basic-object>
      </compound-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">0.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-100.0</basic-object>
      </compound-key>
      <basic-key name="keepAspectRatio">false</basic-key>
      <basic-key name="scale">true</basic-key>
      <basic-key name="paint">#d9d9d9</basic-key>
      <basic-key name="draw-shape">true</basic-key>
      <basic-key name="fill-shape">false</basic-key>
    </style>
    <datasource type="StaticDataSource">
      <compound-object name="value" class="java.awt.geom.Line2D$Float">
        <basic-object name="x2">100.0</basic-object>
        <basic-object name="x1">0.0</basic-object>
        <basic-object name="y2">25.0</basic-object>
        <basic-object name="y1">25.0</basic-object>
      </compound-object>
    </datasource>
  </element>

  <!-- element to display the name from the TableModel -->
  <element name="@anonymous1d" type="text/plain">
    <style>
      <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
        <basic-object name="x">0.0</basic-object>
        <basic-object name="y">8.0</basic-object>
      </compound-key>
      <basic-key name="dynamic_height">false</basic-key>
      <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
        <basic-object name="height" class="java.lang.Double">10.0</basic-object>
        <basic-object name="width" class="java.lang.Double">-50.0</basic-object>
      </compound-key>
      <basic-key name="paint">black</basic-key>
      <basic-key name="alignment">left</basic-key>
      <basic-key name="valignment">bottom</basic-key>
    </style>
    <template references="string-field">
      <basic-object name="nullValue"></basic-object>
      <basic-object name="field">Name</basic-object>
    </template>
  </element>
```

```

<!-- element to display icon from the TableModel -->
<element name="@anonymous" type="image/generic">
  <style>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">-50.0</basic-object>
      <basic-object name="y">1.0</basic-object>
    </compound-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">24.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-30.0</basic-object>
    </compound-key>
    <basic-key name="keepAspectRatio">true</basic-key>
    <basic-key name="dynamic_height">>false</basic-key>
    <basic-key name="scale">>false</basic-key>
    <basic-key name="paint">white</basic-key>
  </style>
  <template references="image-field">
    <basic-object name="field">Icon</basic-object>
  </template>
</element>

<!-- element to display the file size from the TableModel -->
<element name="@anonymous" type="text/plain">
  <style>
    <compound-key name="absolute_pos" class="java.awt.geom.Point2D$Float">
      <basic-object name="x">-80.0</basic-object>
      <basic-object name="y">8.0</basic-object>
    </compound-key>
    <basic-key name="dynamic_height">>false</basic-key>
    <compound-key name="min-size" class="com.jrefinery.report.targets.FloatDimension">
      <basic-object name="height" class="java.lang.Double">10.0</basic-object>
      <basic-object name="width" class="java.lang.Double">-20.0</basic-object>
    </compound-key>
    <basic-key name="paint">black</basic-key>
    <basic-key name="alignment">right</basic-key>
    <basic-key name="valignment">bottom</basic-key>
  </style>
  <template references="number-field">
    <basic-object name="nullValue"></basic-object>
    <basic-object name="format">#,##0</basic-object>
    <basic-object name="field">Size</basic-object>
  </template>
</element>
</itemband>

```

Some points to note:

- unless overridden, text items in the band inherit the font settings specified in the *default-style* element for the band;
- the line element is scaled to cover the full width of the band;
- each of the remaining elements displays an item from the report's `TableModel`. The `field` attribute in the *template* element specifies the name of the column in the `TableModel` from which the data item is read.

### 3.6 Previewing the Report

Once the report template has been constructed, and the data is available, it is simple to create a print preview frame to display the report. The first step is to construct a `JFreeReport` instance from the XML report template file, and link the dataset to this instance. In the example, I use the following code:

```

URL in = getClass().getResource("/com/jrefinery/report/demo/swing-icons.xml");
this.report = parseReport(in);
this.report.setData(this.data);

```

This assumes that the `swing-icons.xml` file is located in the same directory as the `SwingIconsDemo.class` file.

To display the print preview frame for this report:

```

PreviewFrame frame = new PreviewFrame(this.report);
frame.setLargeIconsEnabled(true);
frame.setToolBarFloatable(false);
frame.pack();

```

```
RefineryUtilities.positionFrameRandomly(frame);  
frame.setVisible(true);  
frame.requestFocus();
```

The print preview frame then provides all the standard options to the user, including paging back and forward through the report, zooming, printing, and export to PDF.

## A The GNU Lesser General Public Licence

### A.1 Introduction

JFreeReport is licensed under the terms of the GNU Lesser General Public Licence (LGPL). The full text of this licence is reproduced in this appendix. You should read and understand this licence before using JFreeReport in your own projects.

If you are not familiar with the idea of *free software* and/or *open source software*, you can find out more at the following web-sites:

Organisation:	Description:
The Free Software Foundation	<a href="http://www.fsf.org">http://www.fsf.org</a>
The Open Source Initiative	<a href="http://www.opensource.org">http://www.opensource.org</a>

If you have any questions about the licensing of JFreeReport, please post a message in the JFreeReport forum.

### A.2 The Licence

The following licence has been used for the distribution of the JFreeReport class library:

#### GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the “Lesser” General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a “work based on the library” and a “work that uses the library”. The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## **TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called “this License”). Each licensee is addressed as “you”.

A “library” means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The “Library”, below, refers to any such software library or work which has been distributed under these terms. A “work based on the Library” means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term “modification”.)

“Source code” for a work means the preferred form of the work for making modifications to

it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

\* a) The modified work must itself be a software library.

\* b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

\* c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

\* d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

\* a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

\* b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user’s computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

\* c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

\* d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

\* e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the “work that uses the Library” must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

\* a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

\* b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work. 8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library

under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### **NO WARRANTY**

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### **END OF TERMS AND CONDITIONS**

##### **How to Apply These Terms to Your New Libraries**

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This library is free software; you can redistribute it and/or modify it
under the terms of the GNU Lesser General Public License as published by
the Free Software Foundation; either version 2.1 of the License, or (at
your option) any later version.
```

```
This library is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for
more details.
```

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library  
'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!