The JFreeChart Class Library

Version 1.0.12

Installation Guide

Written by David Gilbert

January 2, 2009

© 2000-2009, 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:
We work hard to make this document as accurate and informative as we can, but cannot guarantee that it is error-free.

Contents

1 Introduction **2**
- 1.1 What is JFreeChart? . . . 2
  - 1.1.1 Overview . . . 2
  - 1.1.2 Features . . . 3
  - 1.1.3 Home Page . . . 3
- 1.2 This Document . . . 4
  - 1.2.1 Versions . . . 4
  - 1.2.2 Disclaimer . . . 4
- 1.3 Acknowledgements . . . 4
- 1.4 Comments and Suggestions . . . 5

2 Sample Charts **6**
- 2.1 Introduction . . . 6
- 2.2 Pie Charts . . . 6
- 2.3 Bar Charts . . . 8
- 2.4 Line Chart . . . 10
- 2.5 XY Plots . . . 11
- 2.6 Time Series Charts . . . 12
- 2.7 Histograms . . . 13
- 2.8 Area Charts . . . 14
- 2.9 Difference Chart . . . 15
- 2.10 Step Chart . . . 16
- 2.11 Gantt Chart . . . 17
- 2.12 Multiple Axis Charts . . . 18
- 2.13 Combined and Overlaid Charts . . . 19
- 2.14 Future Development . . . 20

3 Downloading and Installing JFreeChart **21**
- 3.1 Introduction . . . 21
- 3.2 Download . . . 21
- 3.3 Unpacking the Files . . . 21
  - 3.3.1 Unpacking on Linux/Unix . . . 22
  - 3.3.2 Unpacking on Windows . . . 22
  - 3.3.3 The Files . . . 22
- 3.4 Running the Demonstration Applications . . . 22
- 3.5 Configuring JFreeChart for use in IDEs . . . 23
- 3.6 Compiling the Source . . . 23
- 3.7 Generating the Javadoc Documentation . . . 23

4 The JFreeChart Developer Guide **24**
4.1 Overview . . . 24
4.2 The Guide . . . 24
4.2.1 Site Licences . . . 24
4.3 Demo Application Source Code . . . 24

A Configuring IDEs for JFreeChart **26**
A.1 Introduction . . . 26
A.2 Eclipse . . . 26
A.2.1 Overview . . . 26
A.2.2 Configuration Steps . . . 26
A.2.3 Creating an Eclipse Project that uses JFreeChart . . . 28
A.3 NetBeans . . . 30
A.3.1 Overview . . . 30
A.3.2 Configuration Steps . . . 30
A.3.3 Creating a NetBeans Project that uses JFreeChart . . . 31

B The GNU Lesser General Public Licence **33**
B.1 Introduction . . . 33
B.2 The Licence . . . 33
B.3 Frequently Asked Questions . . . 39
B.3.1 Introduction . . . 39
B.3.2 Questions and Answers . . . 39

Chapter 1

Introduction

1.1 What is JFreeChart?

1.1.1 Overview

JFreeChart is a free chart library for the Java(tm) platform. It is designed for use in applications, applets, servlets and JSP. JFreeChart is distributed with complete source code subject to the terms of the GNU Lesser General Public Licence, which permits JFreeChart to be used in proprietary or free software applications (see Appendix B for details).

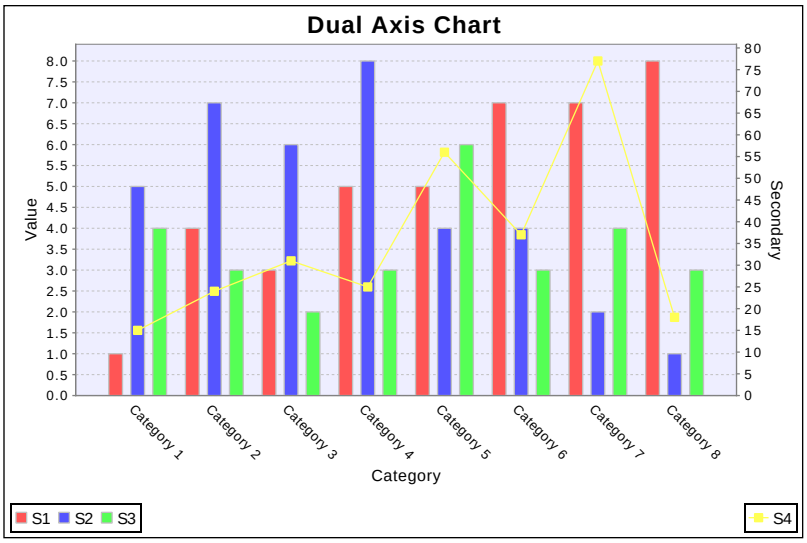


Figure 1.1: A sample chart

Figure 1.1 shows a typical chart created using JFreeChart. Many more examples are shown in later sections of this document.

1.1.2 Features

JFreeChart can generate pie charts, bar charts (regular and stacked, with an optional 3D-effect), line charts, scatter plots, time series charts (including moving averages, high-low-open-close charts and candlestick plots), Gantt charts, meter charts (dial, compass and thermometer), symbol charts, wind plots, combination charts and more.

Additional features include:

- data is accessible from any implementation of the defined interfaces;
- export to PNG and JPEG image file formats (or you can use Java's ImageIO library to export to any format supported by ImageIO);
- export to any format with a `Graphics2D` implementation including:
  - PDF via iText (`http://www.lowagie.com/iText/`);
  - SVG via Batik (`http://xml.apache.org/batik/`);
- tool tips;
- interactive zooming;
- chart mouse events (these can be used for drill-down charts or information pop-ups);
- annotations;
- HTML image map generation;
- works in applications, servlets, JSP (thanks to the Cewolf project[1]) and applets;
- distributed with complete source code subject to the terms of the GNU Lesser General Public License (LGPL);

JFreeChart is written entirely in Java, and should run on any implementation of the Java 2 platform (JDK 1.3.1 or later). It will also work quite well with free runtimes based on GNU Classpath 0.92 or later.[2]

1.1.3 Home Page

The JFreeChart home page can be found at:

`http://www.jfree.org/jfreechart/`

Here you will find all the latest information about JFreeChart, including sample charts, download links, Javadocs, a discussion forum and more.

[1] See `http://cewolf.sourceforge.net` for details.
[2] See `http://www.gnu.org/software/classpath/` for details.

1.2 This Document

1.2.1 Versions

Two versions of this document are available:

- a free version, the "JFreeChart Installation Guide", is available from the JFreeChart home page, and contains chapters up to and including the instructions for installing JFreeChart and running the demo;
- a premium version, the "JFreeChart Developer Guide", is available only to those that have paid for it, and includes additional tutorial chapters and reference documentation for the JFreeChart classes.

If you wish to purchase the latter version, please visit the following site:

`http://www.object-refinery.com/jfreechart/guide.html`

We'd like to thank everyone that has supported JFreeChart in the past by purchasing the JFreeChart Developer Guide!

1.2.2 Disclaimer

Please note that I have put in considerable effort to ensure that the information in this document is up-to-date and accurate, but I cannot guarantee that it does not contain errors. You must use this document *at your own risk* or *not use it at all*.

1.3 Acknowledgements

JFreeChart contains code and ideas from many people. At the risk of missing someone out, I would like to thank the following people for contributing to the project:

Eric Alexander, Richard Atkinson, David Basten, David Berry, Chris Boek, Zoheb Borbora, Anthony Boulestreau, Jeremy Bowman, Daniel Bridenbecker, Nicolas Brodu, Jody Brownell, David Browning, Brian Cabana, Søren Caspersen, Chuanhao Chiu, Brian Cole, Pascal Collet, Martin Cordova, Paolo Cova, Michael Duffy, Don Elliott, Rune Fausk, Jonathan Gabbai, Serge V. Grachov, Daniel Gredler, Hans-Jurgen Greiner, Joao Guilherme Del Valle, Nick Guenther, Aiman Han, Cameron Hayne, Jon Iles, Wolfgang Irler, Sergei Ivanov, Adrian Joubert, Darren Jung, Xun Kang, Bill Kelemen, Norbert Kiesel, Gideon Krause, Pierre-Marie Le Biot, Arnaud Lelievre, Wolfgang Lenhard, David Li, Yan Liu, Tin Luu, Craig MacFarlane, Achilleus Mantzios, Thomas Meier, Aaron Metzger, Jim Moore, Jonathan Nash, Barak Naveh, David M. O'Donnell, Krzysztof Paz, Tomer Peretz, Xavier Poinsard, Andrzej Porebski, Luke Quinane, Viktor Rajewski, Eduardo Ramalho, Michael Rauch, Cameron Riley, Klaus Rheinwald, Dan Rivett, Scott Sams, Michel Santos, Thierry Saura, Andreas Schneider, Jean-Luc Schwab, Bryan Scott, Tobias Self, Mofeed Shahin, Pady Srinivasan, Greg Steckman, Roger Studner, Gerald Struck, Irv Thomae, Eric Thomas, Rich Unger, Daniel van Enckevort, Laurence Vanhelsuwé, Sylvain Vieujot, Jelai Wang, Mark Watson, Alex Weber, Richard West, Matthew Wright, Benoit Xhenseval, Christian W. Zuckschwerdt, Hari and Sam (oldman).

1.4 Comments and Suggestions

If you have any comments or suggestions regarding this document, please send e-mail to:

david.gilbert@object-refinery.com

Chapter 2

Sample Charts

2.1 Introduction

This section shows some sample charts created using JFreeChart. It is intended to give a reasonable overview of the types of charts that JFreeChart can generate. For other examples, please run the demo application included in the JFreeChart distribution:

```
java -jar jfreechart-1.0.12-demo.jar
```

The complete source code for the demo application is available to purchasers of the JFreeChart Developer Guide.[1]

2.2 Pie Charts

JFreeChart can create *pie charts* using any data that conforms to the `PieDataset` interface. Figure 2.1 shows a simple pie chart.

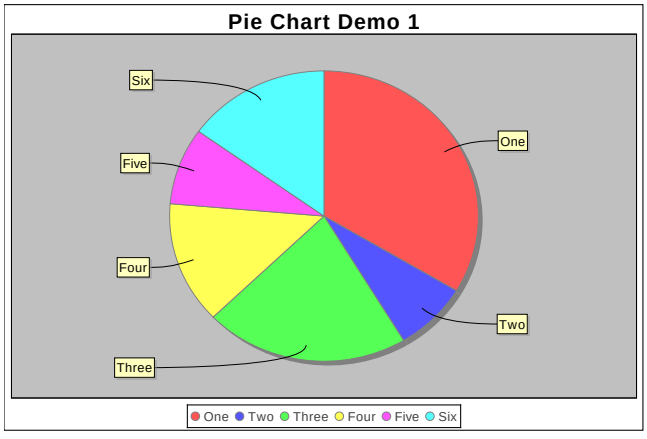


Figure 2.1: A simple pie chart (see `PieChartDemo1.java`)

[1] See `http://www.object-refinery.com/jfreechart/guide.html` for details.

Individual pie sections can be "exploded", as shown in figure 2.2.

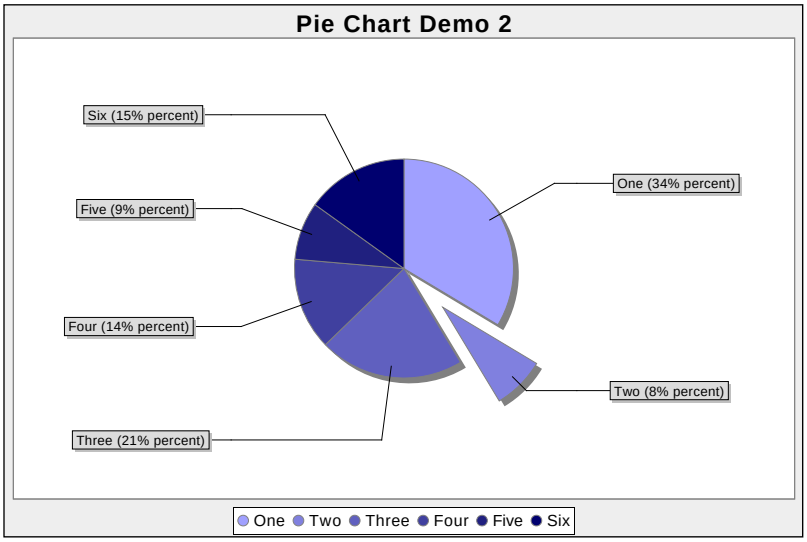


Figure 2.2: A pie chart with an "exploded" section (see `PieChartDemo2.java`)

You can also display pie charts with a 3D effect, as shown in figure 2.3.

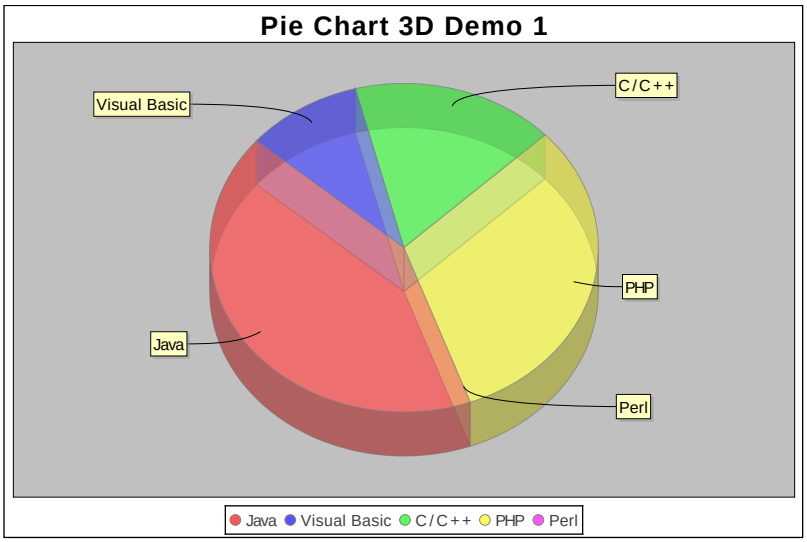


Figure 2.3: A pie chart drawn with a 3D effect (see `PieChart3DDemo1.java`)

At the current time it is *not* possible to explode sections of the 3D pie chart.

2.3 Bar Charts

A range of bar charts can be created with JFreeChart, using any data that conforms to the `CategoryDataset` interface. Figure 2.4 shows a bar chart with a vertical orientation.

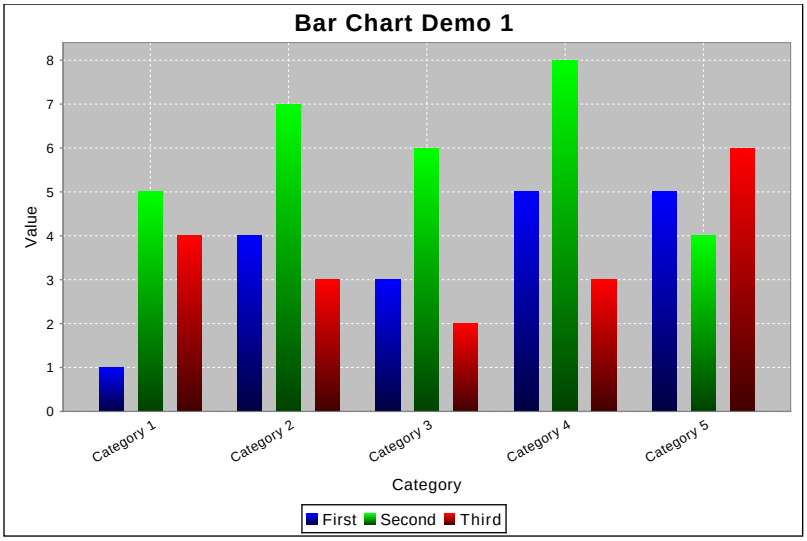


Figure 2.4: A vertical bar chart (see `BarChartDemo1.java`)

Bar charts can be displayed with a 3D effect as shown in figure 2.5.

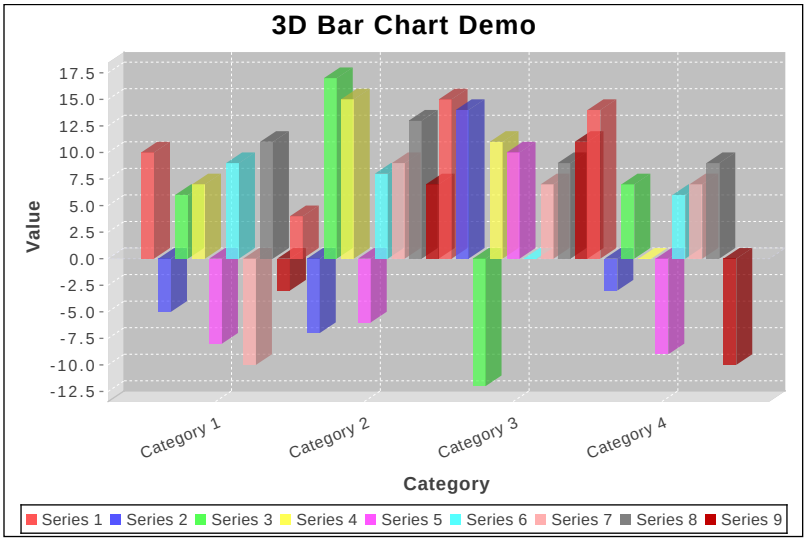


Figure 2.5: A bar chart with 3D effect (see `BarChart3DDemo1.java`)

Another variation, the *waterfall chart*, is shown in figure 2.6.

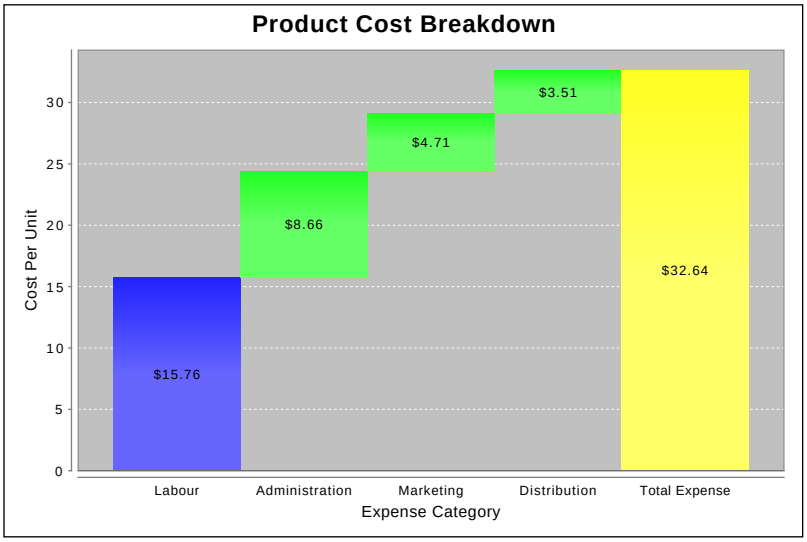


Figure 2.6: A waterfall chart (see `WaterfallChartDemo1.java`)

Bar charts can also be generated from time series data—for example, see figure 2.7:

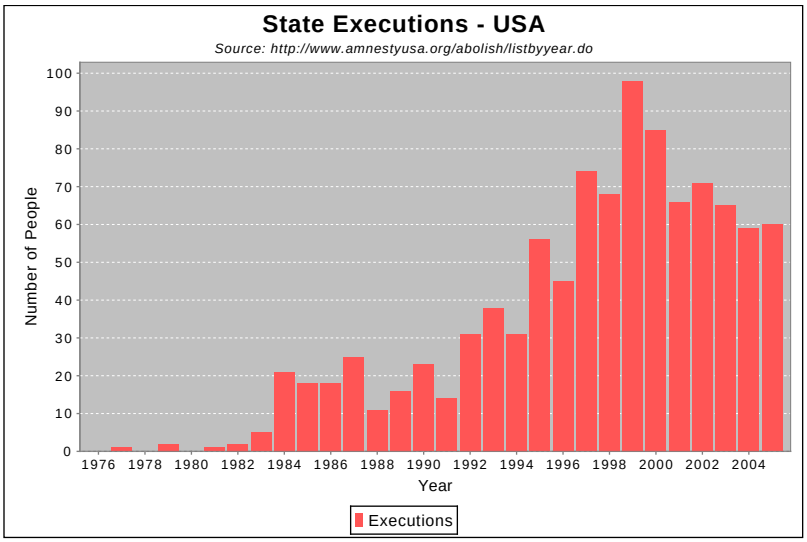


Figure 2.7: An XY bar chart (see `XYBarChartDemo1.java`)

2.4 Line Chart

The *line chart* can be generated using the same `CategoryDataset` that is used for the bar charts—figure 2.8 shows an example.

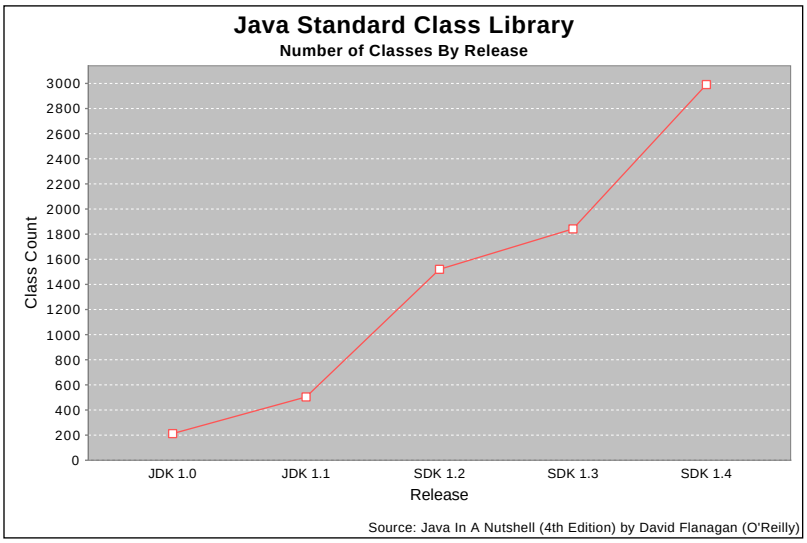


Figure 2.8: A line chart (see `LineChartDemo1.java`)

2.5 XY Plots

A third type of dataset, the `XYDataset`, is used to generate a range of chart types.

The standard *XY plot* has numerical x and y axes. By default, lines are drawn between each data point—see figure 2.9.

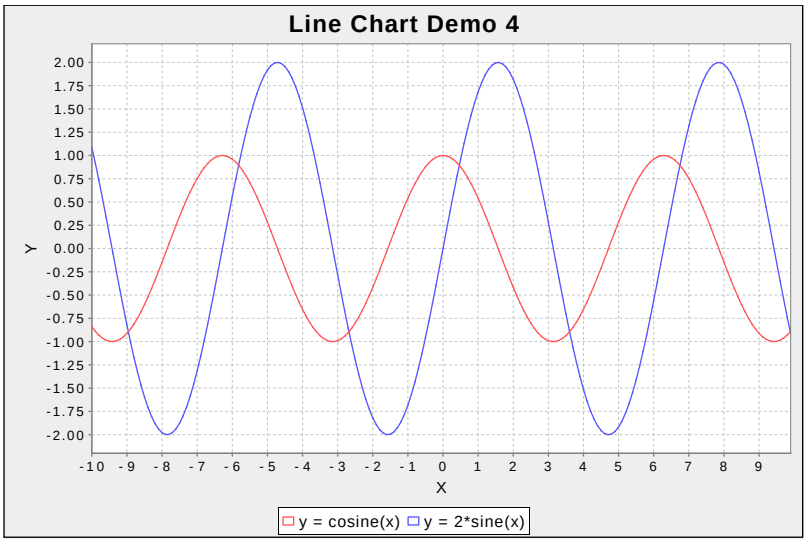


Figure 2.9: A line chart (see `LineChartDemo4.java`)

Scatter plots can be drawn by drawing a shape at each data point, rather than connecting the points with lines—an example is shown in figure 2.10.

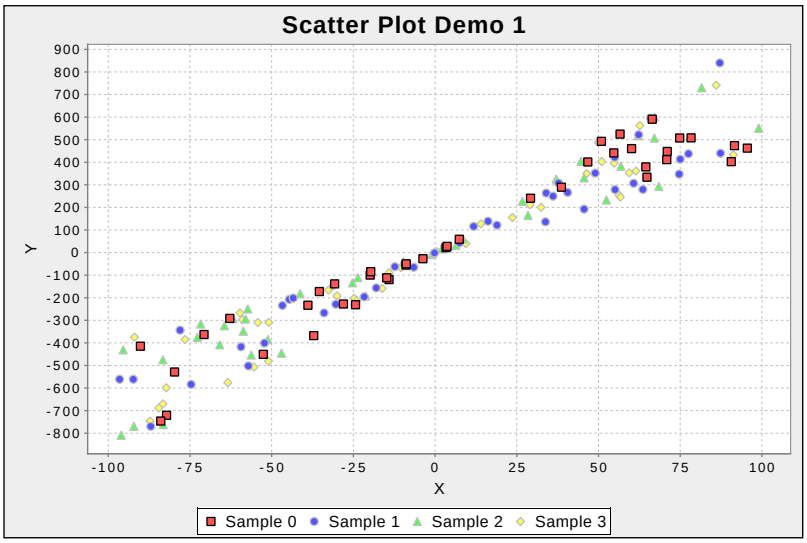


Figure 2.10: A scatter plot (see `ScatterPlotDemo1.java`)

2.6 Time Series Charts

JFreeChart supports *time series charts*, as shown in figure 2.11.

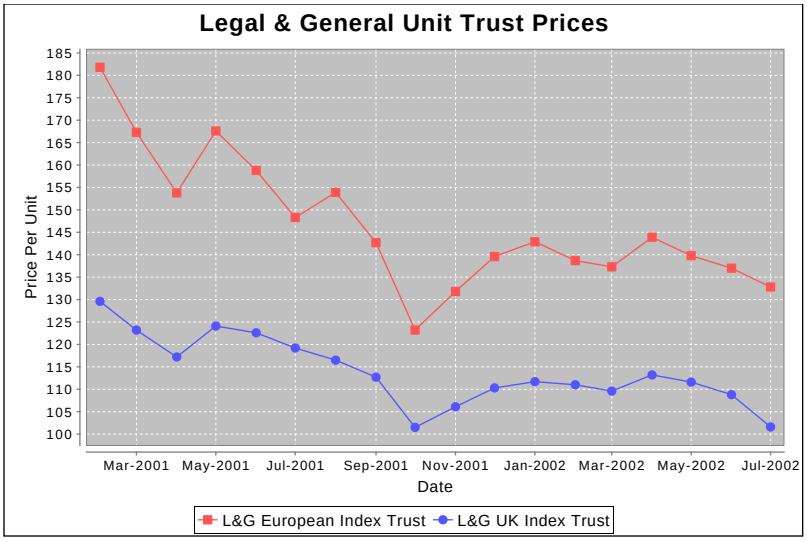


Figure 2.11: A time series chart (see `TimeSeriesDemo1.java`)

It is straightforward to add a moving average line to a time series chart—see figure 2.12 for an example.

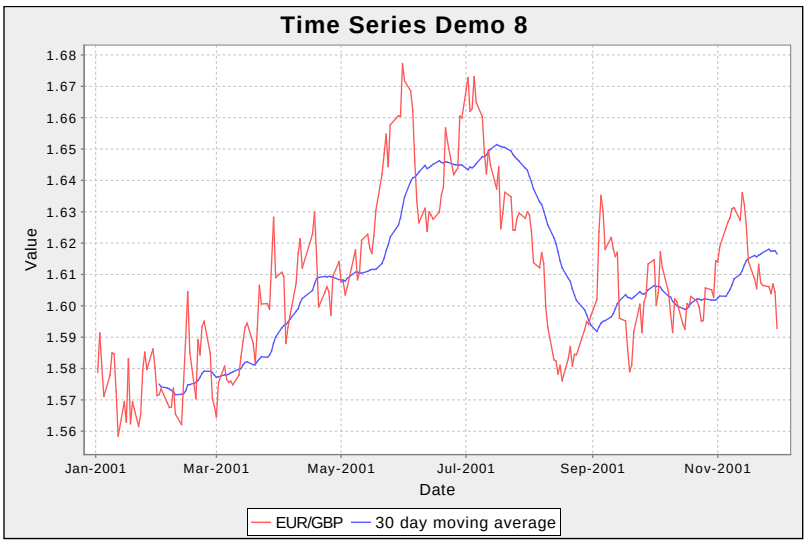


Figure 2.12: A time series chart with a moving average (see `TimeSeriesDemo8.java`)

Using an `OHLCDataset` (an extension of `XYDataset`) you can display *high-low-open-close* data, see figure 2.13 for an example.

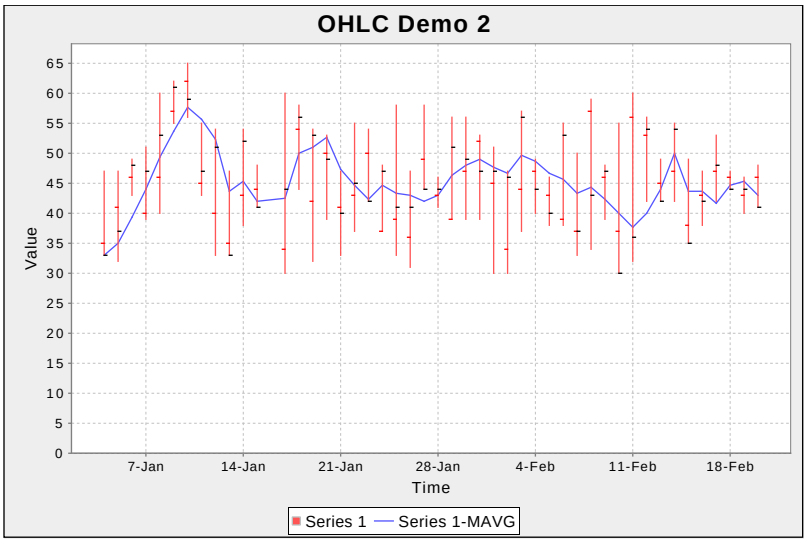


Figure 2.13: A high-low-open-close chart (see `HighLowChartDemo2.java`)

2.7 Histograms

Histograms can be generated using an `IntervalXYDataset` (another extension of `XYDataset`), see figure 2.14 for an example.

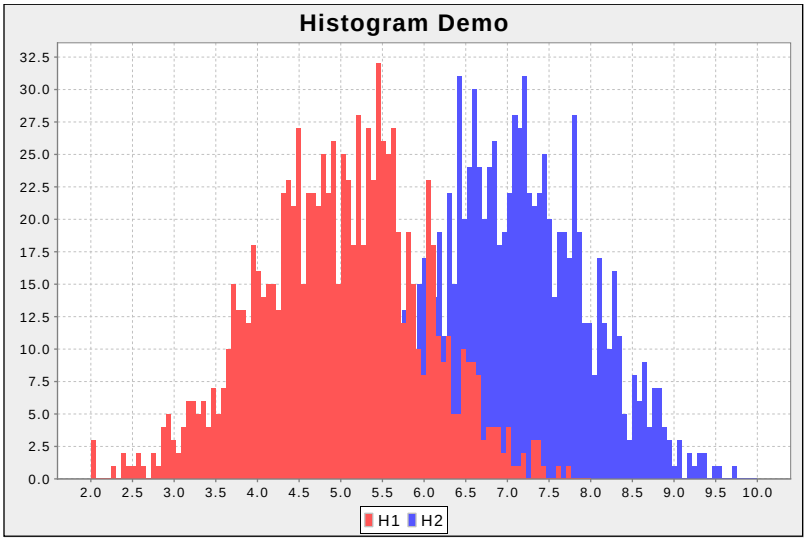


Figure 2.14: A histogram (see `HistogramDemo1.java`)

2.8 Area Charts

You can generate an *area chart* for data in a `CategoryDataset` or an `XYDataset`. Figure 2.15 shows an example.

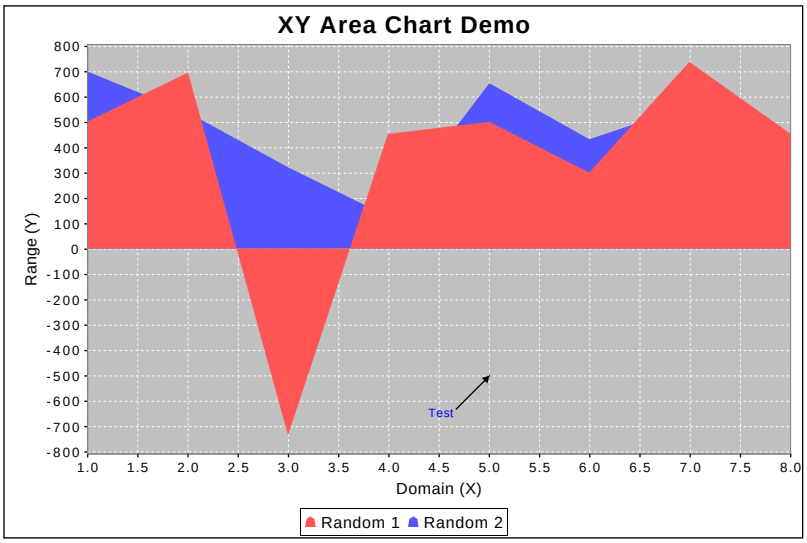


Figure 2.15: An area chart (see `XYAreaChartDemo1.java`)

JFreeChart also supports the creation of *stacked area charts* as shown in figure 2.16.

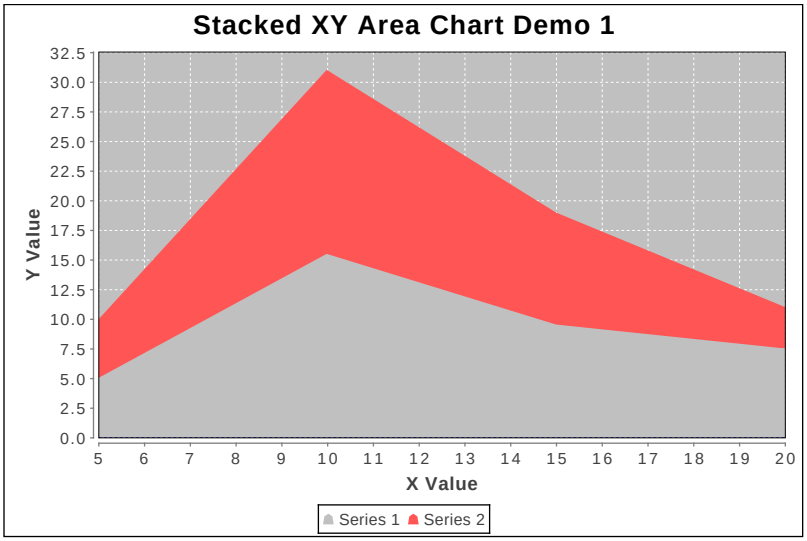


Figure 2.16: A stacked area chart (see `StackedXYAreaChartDemo1.java`)

2.9 Difference Chart

A *difference chart* highlights the difference between two series (see figure 2.17).

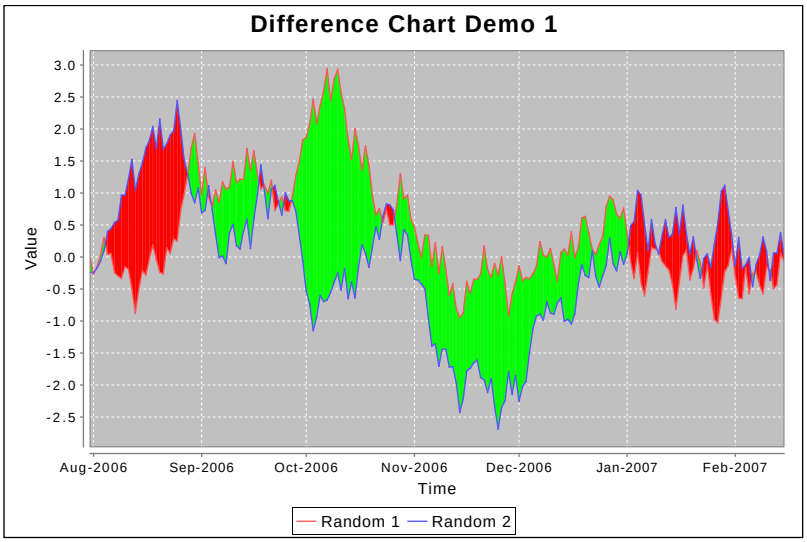


Figure 2.17: A difference chart (see `DifferenceChartDemo1.java`)

A second example, shown in figure 2.18 shows how a date axis can be used for the range values.

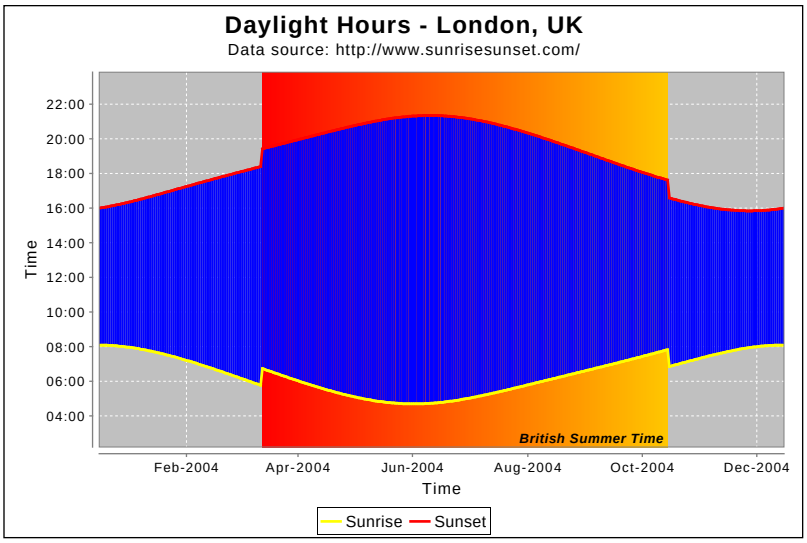


Figure 2.18: A difference chart with times on the range axis (see `DifferenceChartDemo2.java`)

2.10 Step Chart

A *step chart* displays numerical data as a sequence of "steps"—an example is shown in figure 2.19.

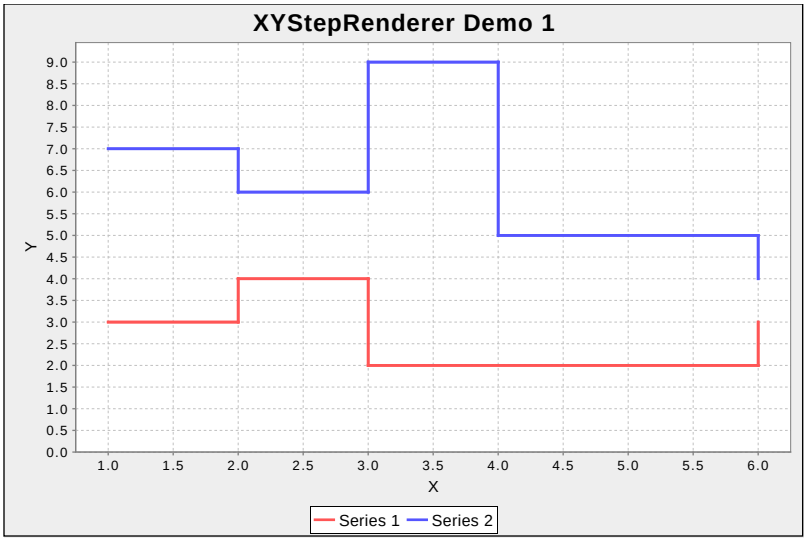


Figure 2.19: A step chart (see `XYStepRendererDemo1.java`)

Step charts are generated from data in an `XYDataset`.

2.11 Gantt Chart

Gantt charts can be generated using data from an `IntervalCategoryDataset`, as shown in figure 2.20.

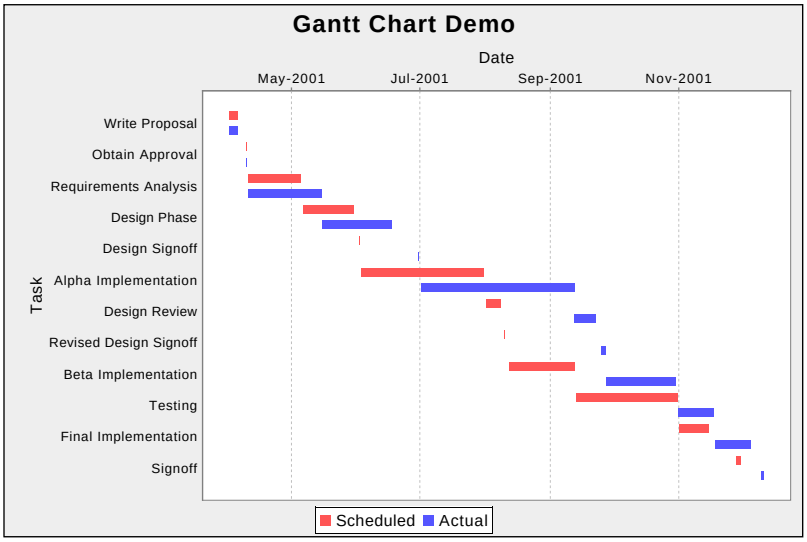


Figure 2.20: A Gantt chart (see `GanttChartDemo1.java`)

Another example, showing subtasks and progress indicators, is shown in figure 2.21.

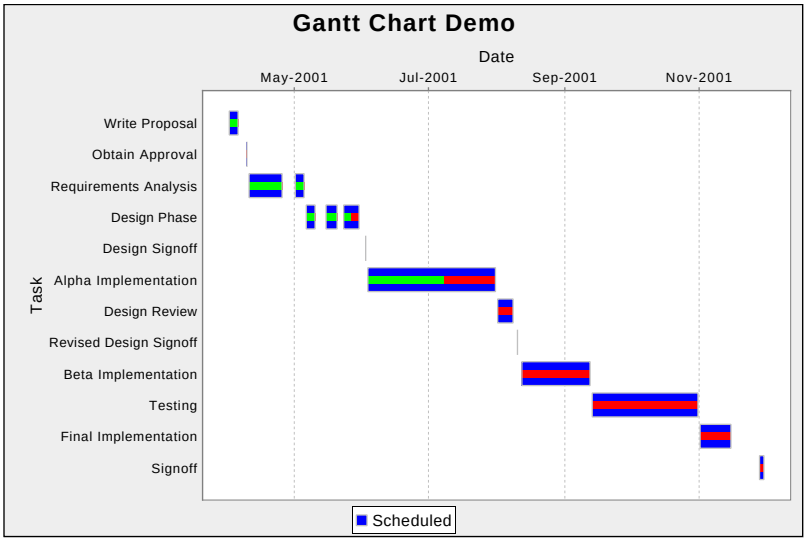


Figure 2.21: A Gantt chart with progress indicators (see `GanttChartDemo2.java`)

2.12 Multiple Axis Charts

JFreeChart has support for charts with multiple axes. Figure 2.22 shows a *price-volume chart* that demonstrates this feature.

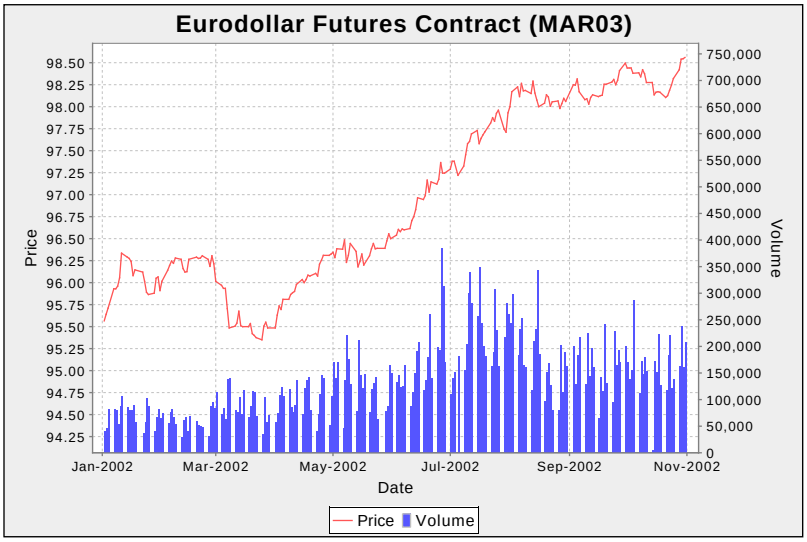


Figure 2.22: A price-volume chart (see `PriceVolumeDemo1.java`)

This feature is supported by the `CategoryPlot` and `XYPlot` classes. Figure 2.23 shows an example with four range axes.

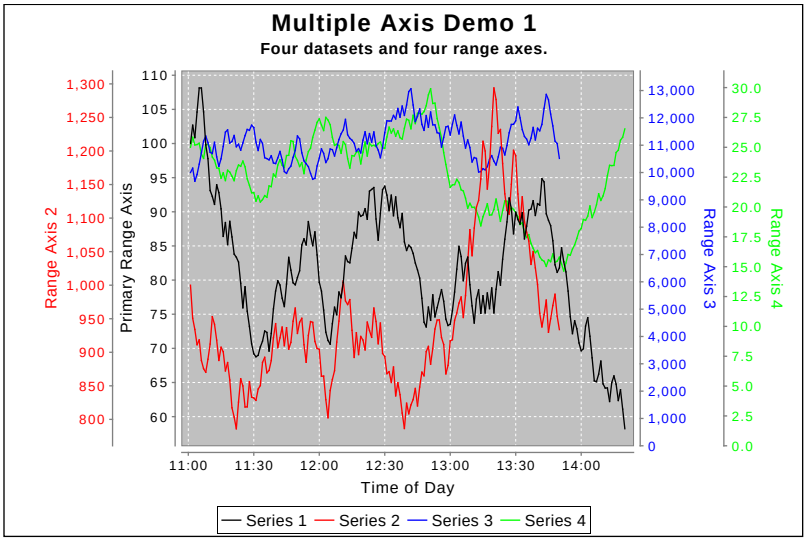


Figure 2.23: A chart with multiple axes (see `MultipleAxisDemo1.java`)

2.13 Combined and Overlaid Charts

JFreeChart supports combined and overlaid charts. Figure 2.24 shows a line chart overlaid on top of a bar chart.

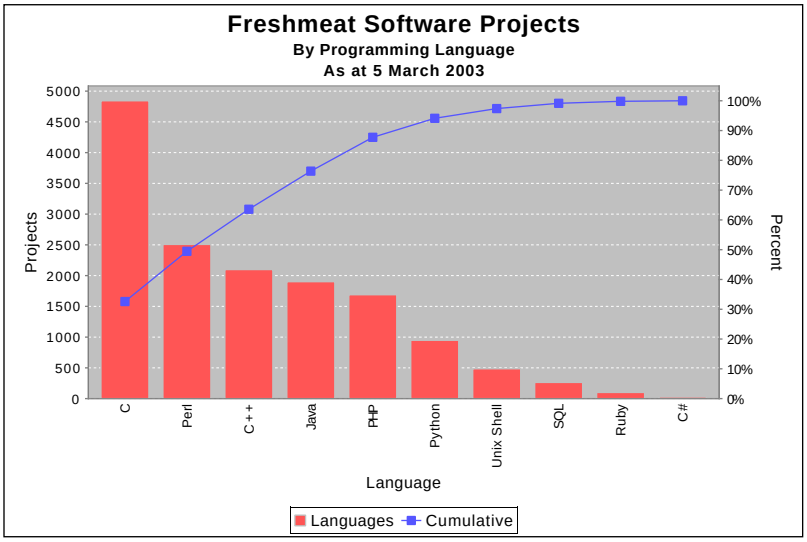


Figure 2.24: An overlaid chart (see `ParetoChartDemo1.java`)

It is possible to combine several charts that share a common domain axis, as shown in figure 2.25.

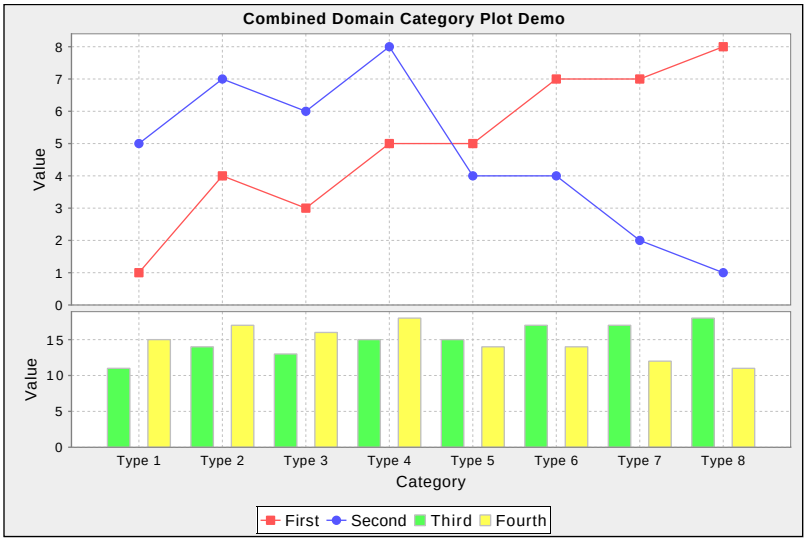


Figure 2.25: A chart with a combined domain (see `CombinedCategoryPlotDemo1.java`)

In a similar way, JFreeChart can combine several charts that share a common range axis, see figure 2.26.

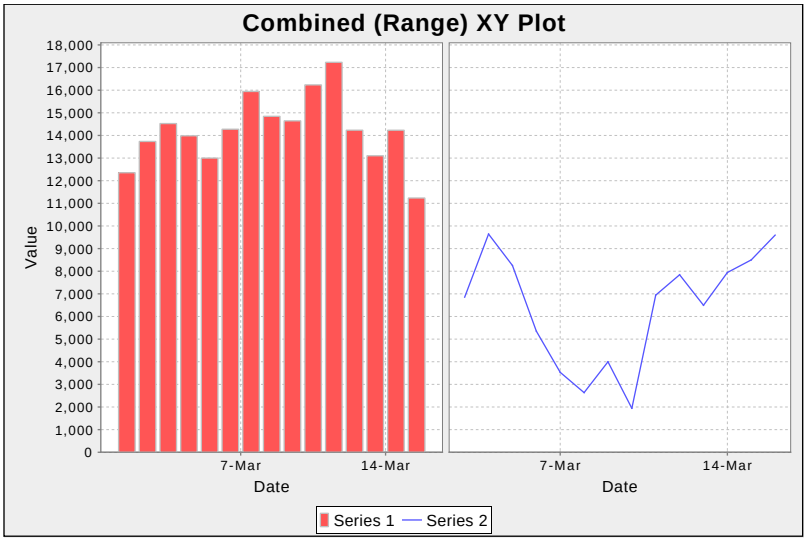


Figure 2.26: A chart with a combined range (see `CombinedXYPlotDemo2.java`)

2.14 Future Development

JFreeChart is *free software*,[2] so anyone can extend it and add new features to it. Already, more than 80 developers from around the world have contributed code back to the JFreeChart project. It is likely that many more chart types will be developed in the future as developers modify JFreeChart to meet their requirements. Check the JFreeChart home page regularly for announcements and other updates:

`http://www.jfree.org/jfreechart/`

And if you would like to contribute code to the project, please join in...

[2] See `http://www.fsf.org`

Chapter 3

Downloading and Installing JFreeChart

3.1 Introduction

This section contains instructions for downloading, unpacking, and (optionally) recompiling JFreeChart. Also included are instructions for running the JFreeChart demonstration application, and generating the Javadoc HTML files from the JFreeChart source code.

3.2 Download

You can download the latest version of JFreeChart from:

`http://www.jfree.org/jfreechart/download/`

There are two versions of the JFreeChart download:

File:	Description:
`jfreechart-1.0.12.tar.gz`	JFreeChart for Linux/Unix.
`jfreechart-1.0.12.zip`	JFreeChart for Windows.

The two files contain the same source code. The main difference is that all the text files in the `zip` download have been recoded to have both carriage return *and* line-feed characters at the end of each line.

JFreeChart uses the JCommon class library (currently version `1.0.15`). The JCommon runtime jar file is included in the JFreeChart download, but if you require the source code (recommended) then you should also download JCommon from:

`http://www.jfree.org/jcommon/`

3.3 Unpacking the Files

After downloading JFreeChart, you need to unpack the files. You should move the download file to a convenient directory—when you unpack JFreeChart, a new subdirectory (`jfreechart-1.0.12`) will be created in the same location as the `zip` or `tar.gz` archive file.

3.3.1 Unpacking on Linux/Unix

To extract the files from the download on Linux/Unix, enter the following command:

```
tar xvzf jfreechart-1.0.12.tar.gz
```

This will extract all the source, run-time and documentation files for JFreeChart into a new directory called `jfreechart-1.0.12`.

3.3.2 Unpacking on Windows

To extract the files from the download on Windows, you can use the `jar` utility. Enter the following command:

```
jar -xvf jfreechart-1.0.12.zip
```

This will extract all the source, run-time and documentation files for JFreeChart into a new directory called `jfreechart-1.0.12`.

3.3.3 The Files

The top-level directory (`jfreechart-1.0.12`) contains the files and directories listed in the following table:

File/Directory:	Description:
`README.txt`	Important information - *read this first!*
`NEWS`	Project news.
`ChangeLog`	A detailed log of changes made to JFreeChart.
`ant`	A directory containing an Ant `build.xml` script. You can use this script to rebuild JFreeChart from the source code included in the distribution.
`checkstyle`	A directory containing several Checkstyle property files. These define the coding conventions used in the JFreeChart source code.
`experimental`	A directory containing source files for classes that are not part of the standard JFreeChart API (yet). We would appreciate feedback on this code. Please note that the API for these classes is subject to change.
`lib`	A directory containing the JFreeChart jar file, and other libraries used by JFreeChart.
`source`	A directory containing the source code for JFreeChart.
`swt`	A directory containing the source code for the experimental SWT code. Please note that the API for these classes is subject to change.
`tests`	A directory containing the source code for the JFreeChart unit tests.
`jfreechart-1.0.12-demo.jar`	A runnable jar file containing demo applications.
`licence-LGPL.txt`	The JFreeChart licence (GNU LGPL).

You should spend some time familiarising yourself with the files included in the download. In particular, you should always read the `README.txt` file.

3.4 Running the Demonstration Applications

A demonstration application is included in the distribution that shows a wide range of charts that can be generated with JFreeChart . To run the demo, type the following command:

```
java -jar jfreechart-1.0.12-demo.jar
```

The source code for the demo application is not included in the JFreeChart distribution, but is available to download separately when you purchase the JFreeChart Developer Guide.[1]

3.5 Configuring JFreeChart for use in IDEs

If, like most developers, you use an integrated development environment (IDE) such as Eclipse or NetBeans for your Java development work, you'll want to configure JFreeChart within that IDE. The procedure for this is IDE-specific—refer to Appendix A for more details.

3.6 Compiling the Source

To recompile the JFreeChart classes, you can use the Ant `build.xml` file included in the distribution. Change to the `ant` directory and type:

```
ant compile
```

This will recompile all the necessary source files and recreate the JFreeChart run-time jar file.

To run the script requires that you have Ant 1.5.1 (or later) installed on your system, to find out more about Ant visit:

```
http://ant.apache.org/
```

It is possible to recompile JFreeChart without using Ant, but there are one or two "gotchas" that you have to take special care to avoid:

- some JFreeChart classes (particularly resource bundles) are not referenced *directly* in the code, and some compilers omit to compile them—this results in runtime errors or problems due to missing class files;
- if you create your own JFreeChart jar file, you need to be sure to include the non-Java files (resource bundle `.properties` files, `gorilla.jpg`, etc.).

In the end, it's simpler to learn Ant and use the script included in the JFreeChart distribution.

3.7 Generating the Javadoc Documentation

The JFreeChart source code contains extensive *Javadoc comments*. You can use the `javadoc` tool to generate HTML documentation files directly from the source code.

To generate the documentation, use the `javadoc` target in the Ant `build.xml` script:

```
ant javadoc
```

This will create a `javadoc` directory containing all the Javadoc HTML files, inside the main `jfreechart-1.0.12` directory.

[1] If you have purchased the guide and you want to download the demo source code, look for the file `jfreechart-1.0.12-demos.zip` on the download page for the JFreeChart Developer Guide.

Chapter 4

The JFreeChart Developer Guide

4.1 Overview

The *JFreeChart Developer Guide* provides extensive documentation for the JFreeChart Class Library. Written by David Gilbert, the principal author of JFreeChart, the guide contains tutorials and reference information that will help you to get the best out of JFreeChart. In addition, the complete source code for the JFreeChart demo application is available for download with the guide

4.2 The Guide

The *JFreeChart Developer Guide* is not free—it is sold by Object Refinery Limited as a means of raising funds for the JFreeChart project. If you would like to support the project financially, please visit the following URL:

`http://www.object-refinery.com/jfreechart/guide.html`

The document is frequently revised and updated—the current version is around 750 pages long. The document is made available via `HTTP` download in Acrobat PDF format (generated in A4 and US letter paper sizes).

Please note that we do NOT ship physical copies of the document.

Note that updates to the JFreeChart Developer Guide are made available free of charge for 1 year after purchase.

4.2.1 Site Licences

Note that there are a couple of site licence options which provide great flexibility for large companies with extensive IT operations, at the same time affording an excellent way to support the ongoing development of JFreeChart. We'd like to say a special "Thank you!" to companies that have already supported us in this way.

4.3 Demo Application Source Code

The source code for the demo application included in the JFreeChart distribution is available to download with the JFreeChart Developer Guide.

In addition, there is:

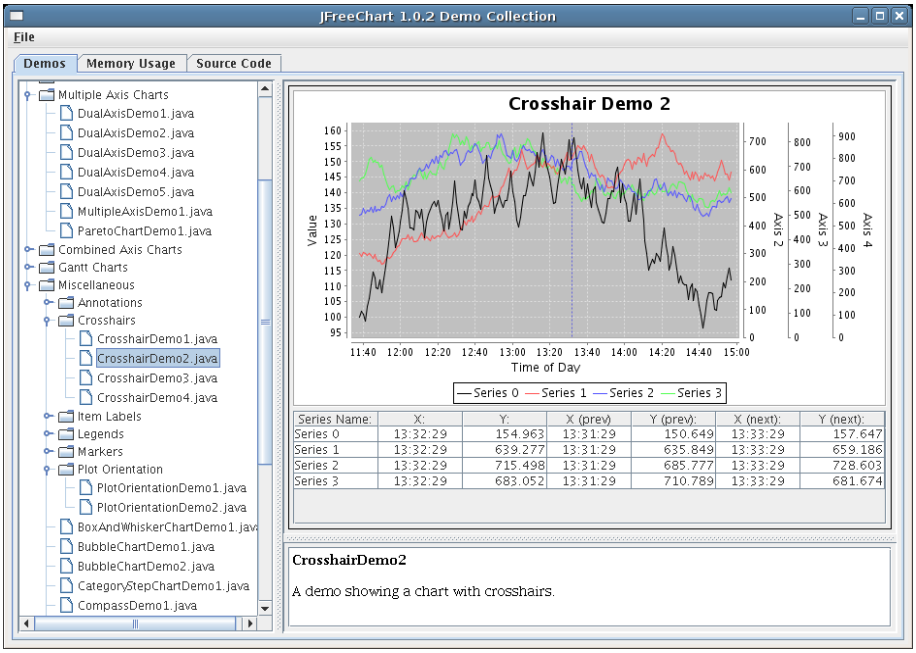


Figure 4.1: The JFreeChart Demo Collection

- a servlet demo, with charts embedded in an HTML page;
- several JDBC demos, where charts are generated using data from a relational database;
- demos showing how to capture chart mouse events;

The servlet and JDBC demos are described in the JFreeChart Developer Guide, including all the steps required for configuration.[1]

[1] Using Tomcat for the servlet demo and PostgreSQL for the JDBC demos.

Appendix A

Configuring IDEs for JFreeChart

A.1 Introduction

There are a number of IDEs (integrated development environments) that developers use when working on Java programs. In this section, I describe how to configure some popular IDEs to use JFreeChart.[1] Specifically, I'll cover:

- Eclipse (version 3.2);
- NetBeans (version 5.5);

In the future I'll add configuration descriptions for other IDEs.[2]

A.2 Eclipse

A.2.1 Overview

Eclipse is a free IDE originally developed by IBM, but now managed by the Eclipse Foundation:

```
http://www.eclipse.org/
```

In Eclipse, third party libraries are configured as "user libraries". In this section, I'll describe how to set up JFreeChart and JCommon as user libraries in Eclipse 3.2. This makes it straightforward to include JFreeChart and JCommon as dependencies in your application(s), with Eclipse automatically handling features like code-completion, Javadoc popups, stepping through the JFreeChart/JCommon sources during debugging, and more.

A.2.2 Configuration Steps

To begin with, you need to download the JFreeChart and JCommon distributions, unpack them on your local machine, and generate the API documentation. The following steps are necessary:

1. Download the latest version of the JCommon class library:

   ```
   http://www.jfree.org/jcommon/
   ```

[1] Notes that this section is concerned with *using* JFreeChart as a library. If you intend to *modify* the JFreeChart sources, you'll want to configure JFreeChart as a project within your IDE.

[2] At least those I can get access to.

...and unpack it to a directory on your computer (almost anywhere is fine).

2. From the `ant` subdirectory of the just-unpacked JCommon, run `ant javadoc` to generate the Javadocs locally. If you are unfamiliar with Ant, you can skip this step, but then Eclipse won't be able to show you the Javadoc popups for JCommon.

3. Download the latest version of the JFreeChart class library:

```
http://www.jfree.org/jfreechart/
```

...and unpack it to a directory on your computer (again, almost anywhere is fine).

4. From the `ant` subdirectory of the just-unpacked JFreeChart, run `ant javadoc` to generate the Javadocs locally. As with step 2, you can skip this step, but then you'll be missing the API documentation.

Now, launch Eclipse, and carry out the following steps to configure JFreeChart and JCommon as user libraries:

5. In Eclipse, select `Preferences...` from the `Window` menu, then choose the `Java -> Build Path -> User Libraries` node in the tree—you should see the dialog shown in figure A.1.

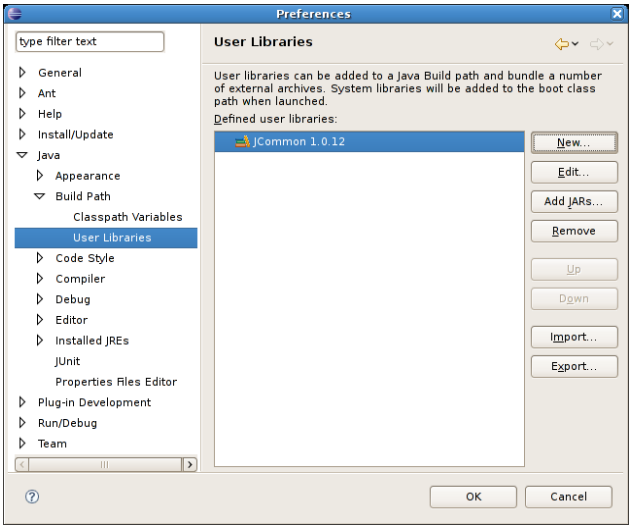


Figure A.1: Eclipse User Libraries Dialog.

6. Click on the `New...` button and enter `JCommon 1.0.12` as the name for a new user library.

7. Ensure that the `JCommon 1.0.12` item is selected in the list, then click the `Add JARs...` button and select the `jcommon-1.0.12.jar` file from the JCommon directory created back in step 1.

8. Double-click the item that says "`Source attachment: (None)`", then click the `External folder...` button, then select the `source` directory for JCommon.

9. Double-click the item that says "`Javadoc location: (None)`", then click the `Browse...` button, then select the `javadoc` directory from JCommon (see step 2).

10. Click on the `New...` button and enter `JFreeChart 1.0.7` as the name for a new user library.

11. Ensure that the `JFreeChart 1.0.7` item is selected in the list, then click on the `Add JARs...` button and select the `jfreechart-1.0.7.jar` file from the JFreeChart directory (see step 3).

12. Double-click the item that says "`Source attachment: (None)`", then click the `External folder...` button, then select the `source` directory for JFreeChart.

13. Double-click the item that says "`Javadoc location: (None)`", then click the `Browse...` button, then select the `javadoc` directory from JFreeChart (see step 4).

At this point, you have completed the configuration of the user libraries—you should have something that looks like figure A.2.

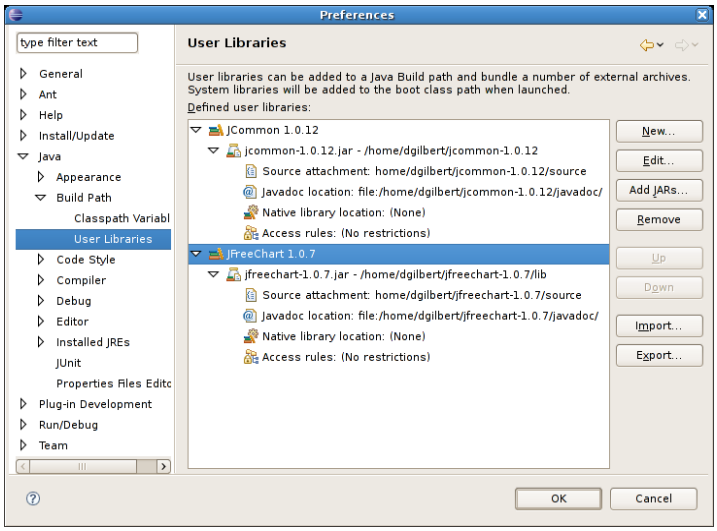


Figure A.2: The Configured User Libraries.

The next section shows how to create a new project in Eclipse that depends on these libraries.

A.2.3 Creating an Eclipse Project that uses JFreeChart

Now that JFreeChart and JCommon are configured as user libraries, it is straightforward to develop an application that uses these libraries:

1. In Eclipse, select `New -> Project...` from the `File` menu, select `Java Project` from the list and click the `Next` button.

2. Enter `MyAppThatUsesJFreeChart` as the project name and click the `Finish` button.

3. Right-click on the project in the `Package Explorer` then select `Properties` from the pop-up menu. In the properties window—see figure A.3—click on the `Add Library...` button and select both the JCommon and JFreeChart libraries. Click `OK`.

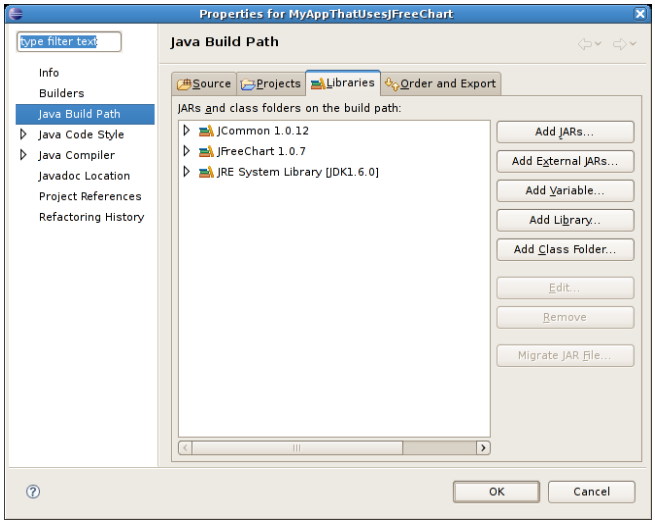


Figure A.3: The completed libraries.

4. Create a new source file (`First.java`) in the project, and copy and paste the following small application:

```
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartFrame;
import org.jfree.chart.JFreeChart;
import org.jfree.data.general.DefaultPieDataset;

/**
 * A simple introduction to using JFreeChart.  This demo is described in the
 * JFreeChart Developer Guide.
 */
public class First {

    /**
     * The starting point for the demo.
     *
     * @param args  ignored.
     */
    public static void main(String[] args) {

        // create a dataset...
        DefaultPieDataset data = new DefaultPieDataset();
        data.setValue("Category 1", 43.2);
        data.setValue("Category 2", 27.9);
        data.setValue("Category 3", 79.5);

        // create a chart...
        JFreeChart chart = ChartFactory.createPieChart(
            "Sample Pie Chart",
            data,
            true,    // legend?
            true,    // tooltips?
            false    // URLs?
        );

        // create and display a frame...
```

```
            ChartFrame frame = new ChartFrame("First", chart);
            frame.pack();
            frame.setVisible(true);

        }

    }
```

5. Compile and run the application. Notice how you can browse the JFreeChart/JCommon source files and step through the code while debugging.

That's all there is to it!

A.3 NetBeans

A.3.1 Overview

NetBeans is a free IDE developed by Sun Microsystems:

`http://www.netbeans.org/`

In NetBeans, third party libraries are configured using the "Library Manager". In this section, I'll describe how to set up JFreeChart and JCommon within the Library Manager in NetBeans version 5.5. This makes it straightforward to include JFreeChart and JCommon as dependencies in your application(s), with NetBeans automatically handling features like code completion, Javadoc popups, stepping through the JFreeChart/JCommon sources during debugging, and more.

A.3.2 Configuration Steps

To begin with, you need to download the JFreeChart and JCommon distributions, unpack them on your local machine, and generate the API documentation. The following steps are necessary:

1. Download the latest version of the JCommon class library:

   `http://www.jfree.org/jcommon/`

   ...and unpack it to a directory on your computer (almost anywhere is fine).

2. From the `ant` subdirectory of the just-unpacked JCommon, run `ant javadoc` to generate the Javadocs locally. If you are unfamiliar with Ant, you can skip this step, but then NetBeans won't be able to show you the Javadoc popups for JCommon.

3. Download the latest version of the JFreeChart class library:

   `http://www.jfree.org/jfreechart/`

   ...and unpack it to a directory on your computer (again, almost anywhere is fine).

4. From the `ant` subdirectory of the just-unpacked JFreeChart, run `ant javadoc` to generate the Javadocs locally. As with step 2, you can skip this step, but then you'll be missing the API documentation.

Now, launch NetBeans, and carry out the following steps to configure JFreeChart and JCommon as user libraries:

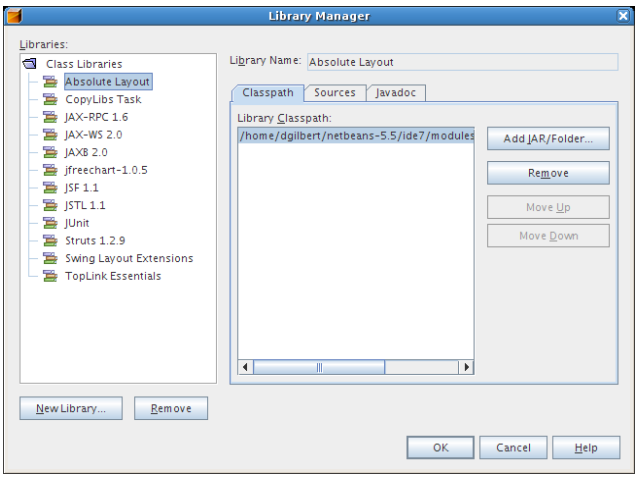


Figure A.4: The Library Manager.

5. In NetBeans, select the `Library Manager` item from the `Tools` menu—you should see the dialog shown in figure A.4.

6. Click on the `New Library...` button and enter `JCommon-1.0.12` as the library name.

7. With the `Classpath` tab selected, click on the `Add JAR/Folder...` button and select the `jcommon-1.0.12.jar` file from the JCommon directory created back in step 1.

8. With the `Sources` tab selected, click on the `Add JAR/Folder...` button and select the `source` directory for JCommon.

9. With the `Javadoc` tab selected, click on the `Add ZIP/Folder...` button and select the `javadoc` directory for JCommon (refer to step 2).

10. Click on the `New Library...` button and enter `JFreeChart-1.0.7` as the library name.

11. With the `Classpath` tab selected, click on the `Add JAR/Folder...` button and select the `jfreechart-1.0.7.jar` file from the JFreeChart directory created back in step 3.

12. With the `Sources` tab selected, click on the `Add JAR/Folder...` button and select the `source` directory for JFreeChart.

13. With the `Javadoc` tab selected, click on the `Add ZIP/Folder...` button and select the `javadoc` directory for JFreeChart (refer to step 4).

At this point, you have complete the configuration of the libraries. The next section shows how to create a new project in NetBeans that depends on these libraries.

A.3.3 Creating a NetBeans Project that uses JFreeChart

Now that JFreeChart and JCommon are configured as libraries in NetBeans, it is straightforward to develop an application that uses these libraries:

1. In NetBeans, select `New Project...` from the `File` menu, select `General/Java Application`, and click the `Next` button.

2. Enter `MyAppThatUsesJFreeChart` as the project name, and click the `Finish` button.

3. In the `Projects` pane, you'll see a `Libraries` node in the project. Right-click on this node, select `Add Library...` and select the JFreeChart and JCommon libraries.

4. NetBeans has already created a `Main.java` source file—copy and paste the following code into the main method of this source file:

```
public static void main(String[] args) {

    // create a dataset...
    DefaultPieDataset data = new DefaultPieDataset();
    data.setValue("Category 1", 43.2);
    data.setValue("Category 2", 27.9);
    data.setValue("Category 3", 79.5);

    // create a chart...
    JFreeChart chart = ChartFactory.createPieChart(
        "Sample Pie Chart",
        data,
        true,     // legend?
        true,     // tooltips?
        false     // URLs?
    );

    // create and display a frame...
    ChartFrame frame = new ChartFrame("First", chart);
    frame.pack();
    frame.setVisible(true);
}
```

5. Select `Fix Imports` from the `Source` menu, then compile and run the application. Notice how you can browse the JFreeChart/JCommon source files and step through the code while debugging.

That's all there is to it!

Appendix B

The GNU Lesser General Public Licence

B.1 Introduction

JFreeChart 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 JFreeChart in your own projects.

If you are not familiar with the idea of *free software*, you can find out more at the Free Software Foundation's web site:

`http://www.fsf.org`

Please send e-mail to `david.gilbert@object-refinery.com` if you have any questions about the licensing of JFreeChart (but please read section B.3 first).

B.2 The Licence

The following licence has been used for the distribution of the JFreeChart 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!

B.3 Frequently Asked Questions

B.3.1 Introduction

Some of the most frequently asked questions about JFreeChart concern the licence. I've published this FAQ to help developers understand my choice of licence for JFreeChart. If anything is unclear, or technically incorrect, please e-mail me (`david.gilbert@object-refinery.com`) and I will try to improve the text.

B.3.2 Questions and Answers

1. "Can I incorporate JFreeChart into a proprietary (closed-source) application?"

Yes, the GNU Lesser General Public Licence (LGPL) is specifically designed to allow this.

2. "Do I have to pay a licence fee to use JFreeChart?"

No, JFreeChart is free software. You are not required to pay a fee to use JFreeChart. All that we ask is that you comply with the terms of the licence, which (for most developers) is not very difficult.

If you want to make a financial contribution to the JFreeChart project, you can buy a copy of the JFreeChart Developer Guide from Object Refinery Limited. This is appreciated, but not required.

3. "If I use JFreeChart, do I have to release the source code for my application under the terms of the LGPL?"

No, you can choose whatever licence you wish for your software. But when you distribute your application, you must include the complete source code for JFreeChart—including any changes you make to it—under the terms of the LGPL. Your users end up with the same rights in relation to JFreeChart as you have been granted under the LGPL.

4. "My users will never look at the source code, and if they did, they wouldn't know what to do with it...why do I have to give it to them?"

The important point is that your users have access to the source code—whether or not they choose to use it is up to them. Bear in mind that non-technical users *can* make use of the source code by hiring someone else to work on it for them.

5. "What are the steps I must follow to release software that incorporates JFreeChart?"

The steps are listed in the licence (see section 6 especially). The most important things are:

- include a notice in your software that it uses the JFreeChart class library, and that the library is covered by the LGPL;
- include a copy of the LGPL so your users understand that JFreeChart is distributed WITHOUT WARRANTY, and the rights that they have under the licence;
- include the complete source code for the version of the library that you are distributing (or a written offer to supply it on demand);

6. "I want to display the JFreeChart copyright notice, what form should it take?"

Try this:

> *This software incorporates JFreeChart, (C)opyright 2000-2007 by Object Refinery Limited and Contributors.*

7. "The LGPL is unnecessarily complicated!"

OK, that's not a question, but the point has been raised by a few developers.

Yes, the LGPL is complicated, but only out of necessity. The complexity is mostly related to the difficulty of defining (in precise legal terms) the relationship between a free software library and a proprietary application that uses the library.

A useful first step towards understanding the LGPL is to read the GNU General Public Licence (GPL). It is a much simpler licence, because it does not allow free software to be combined with non-free (or proprietary) software. The LGPL is a superset of the GPL (you are free to switch from the LGPL to the GPL at any time), but slightly more "relaxed" in that it allows you to combine free and non-free software.

A final note, some of the terminology in the LGPL is easier to understand if you keep in mind that the licence was originally developed with statically-linked C programs in mind. Ensuring that it is possible to relink a modified free library with a non-free application, adds significant complexity to the licence. For Java libraries, where code is dynamically linked, modifying and rebuilding a free library for use with a non-free application needn't be such a big issue, particularly if the free library resides in its own jar file.

8. "Who developed the licence?"

The licence was developed by the Free Software Foundation and has been adopted by many thousands of free software projects. You can find out more information at the Free Software Foundation website:

```
http://www.fsf.org
```

The Free Software Foundation performs important work, please consider supporting them financially.

9. "Have you considered releasing JFreeChart under a different licence, such as an "Apache-style" licence?"

Yes, a range of licences was considered for JFreeChart, but now that the choice has been made there are no plans to change the licence in the future.

A publication by Bruce Perens was especially helpful in comparing the available licences:

```
http://www.oreilly.com/catalog/opensources/book/perens.html
```

In the end, the LGPL was chosen because it is the closest fit in terms of my goals for JFreeChart. It is not a perfect licence, but there is nothing else that comes close (except the GPL) in terms of protecting the freedom of JFreeChart for everyone to use. Also, the LGPL is very widely used, and many developers are already familiar with its requirements.

Some other open source licences (for example the Apache Software Licence) allow open source software to be packaged and redistributed without source code. These licences offer more convenience to developers (especially in large companies) than the LGPL, but they allow a path from open source software to closed source software, which is not something I want to allow for JFreeChart.