CHAPTER

1

Leveraging Open Source Software

Everything that is really great and inspiring is created by the individual who can labor in freedom.

Albert Einstein

Historically, enterprise software has been expensive and proprietary. Furthermore, if you needed to modify the software to suit your specific business needs, you had to contract with the vendor to make changes. Even worse, if there were errors, you were dependent on the vendor to address them in a timely fashion. Open source software has changed all of that.

This chapter examines the question "What is *open source software*?" To answer the question, we briefly describe some of the history of and concepts behind open source software. We also discuss the nature of open source software and the licenses that govern its use. Finally, we provide an overview of the major open source projects supporting Java development.

What Is Open Source Software?

Questions and misunderstandings about open source software and what it is abound. Many people associate open source software with its being free and think that any software that is given away is open source. Although open source software is free, just being free does not make the software open source. To be open source software, the software must meet certain criteria. In general, these can be stated as:

- Open source software allows free redistribution of source code and binaries.
- Open source software provides source code.
- Open source software allows modification and redistribution of the modified source code.

The formal definition of open source software is that open source software is software that is copyrighted and licensed under a license that meets the criteria outlined in the Open Source Definition, which can be found at the following site:

```
http://www.opensource.org/docs/definition.php
```

An alternative definition, known as the Debian Free Software Guidelines, can be found at the following site:

```
http://www.debian.org/social_contract.html#guidelines
```

Origins of the Open Source Movement

Conceptually, the open source software movement began with the GNU project started by Richard Stallman in 1983. GNU was a project established to create a Unix-compatible system of software that would be completely free, to be copied and given away, and that would provide access to all of the source code. Linux exists today mainly because of the tools created for the GNU project. The statements from the GNU Manifesto below describe Richard Stallman's view of Free Software.

> *GNU is not in the public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free.*

The Free Software Foundation was formed to aid in the coordination, development, and maintenance of the GNU project. The Free Software Foundation (FSF) identifies their products as "free software." However, when the FSF refers to "free software" they state that "free" does not refer to "price" but rather to freedom. The FSF compares free software to "free speech" (as opposed to "free beer").

Prior to GNU and FSF, free software was generally "public domain." The FSF addressed the issues of public domain software by copyrighting all of their software. Along with the copyright is a license allowing the use of the software but including certain restrictions. The GNU General Public License,

or GPL, enforces the goals of the FSF by ensuring that the software can be copied, distributed, modified, and extended, but requires that the software always be free and include source code (which the FSF sees as a condition of software freedom). Applying the GPL is often referred to as "copylefting."

The Open Source Software Movement Today

Free software was firmly established in 1998 when the term "open source" was coined. A group of people actively involved in working with Linux and other "free" software projects determined that a new name was needed that could be used to identify all software projects that provided access to source code without restrictions on changes or redistribution. They wanted to include the broader spectrum of free software, not just those programs covered under the GPL. They also wanted to make these software projects more palatable for businesses in the software industry. The FSF with its philosophy and rhetoric about free software had gone a long way toward endearing some programmers, but had alienated many, including businesses. The open source movement grew out of a desire to promote the concepts of free software but to avoid its confrontational past. They wanted to help businesses understand how this free software could be used to their benefit. This new name, open source software, quickly spread and was adopted by many. The Open Source Initiative (OSI) was formed to promote open source software and the concepts behind it. The OSI Web site can be found at the following URL:

```
http://www.opensource.org
```

FSF elected, however, to continue using the term "free software" to describe the programs that are covered by GPL-compatible licenses.

Open Source Licenses

In the previous discussion, we mentioned the GNU Public License, or GPL. The GPL works because of the power of the copyright laws. The copyright establishes ownership. The holder of the copyright is the owner of the software. The owner of the software is able to establish the rules for allowable use of their property by others. These rules are stated in a license. A license is a legal document that gives permission to do specified things or defines restrictions on what can be done.

In the following sections, we look at several of the most popular open source software licenses. Many open source programs share or reuse the licenses from other projects. The GPL for example is used by many projects that are not a part of the GNU project or other FSF efforts. For each license, we try to cover

the most important features of the license and discuss any issues that may arise when combining products with different licenses.

Many of the licenses have similarities. Almost all of the licenses have a provision that states that the software is not covered under any warranties and that use of the software constitutes acceptance of the license. Some licenses also require that changes to the code be clearly marked and distributed separately to avoid confusion with base distributions.

All of the licenses that we discuss in the following sections are recognized by the OSI as open source licenses.

NOTE We are not lawyers. The discussion in this chapter is based on frequently asked questions (FAQ) documents available online and is not intended for use as legal advice. Licenses are, after all, legal documents.

NOTE There are many more open source licenses than those covered here. We have tried to focus on those licenses that we encounter most frequently with the software used throughout this book. Any specific piece of software may have a unique license of its own or a different variant from those covered in this chapter. You should always read and understand the software license before you commit to using a piece of software for any project or endeavor, even if it is an open source license.

Apache

The Apache software license was created originally for the Apache Web server but now applies to all projects hosted under the Apache umbrella. Basically, the license does not have any restrictions on the redistribution of source code or binaries other than the inclusion of a required disclaimer statement. It does require that if Apache software is used in a product the use must be acknowledged in the documentation for the product. The FSF recognizes the Apache license as a "free" license but also states that the Apache license is not compatible with GPL. FSF does not provide a specific reason why the Apache license is not compatible. On the other hand, the Apache License FAQ states that the Apache license is compatible with GPL as far as they can tell and states that FSF has not provided them with an explanation for why it is not GPL compatible.

Artistic

The Artistic License is commonly associated with the Perl programming language. The license is designed to provide the copyright holder with a form of

"artistic control" over the development of the software. It allows redistribution of the software in original or modified form for free or for a fee. It does provide restrictions on how modified versions can be redistributed. These restrictions cover several common scenarios and are mostly oriented toward making certain that modified versions are not confused with the original version. The FSF recognizes the Artistic License as a free software license that is compatible with the GPL.

BSD

The BSD License was created for the University of California at Berkeley. It is a license template. The template is used to generate a license by substituting values for *OWNER*, *ORGANIZATION*, and *YEAR* with ones that are appropriate for the copyright holder. The license provides only three restrictions summarized here:

- Binary distributions must reproduce the copyright notice, the license conditions, and the disclaimer.
- Source distributions must reproduce the copyright notice, the license conditions, and the disclaimer.
- The names of the organization or contributors cannot be used for endorsements without permission.

The FSF recognizes the BSD license as a free software license that is compatible with the GPL.

GNU General Public License (GPL)

The GPL is the primary license of the Free Software Foundation and the GNU project. Many other projects have also adopted the GPL as their standard license. The primary features of the GPL are:

- The program can be copied and redistributed as long as copyright notices, disclaimers, and references to the license remain intact.
- Changes to the program are allowed as long as the changes are identified.
- The modified program must also be licensed under GPL and be freely distributed.
- Copyright notice, disclaimers, and references to the license need to be displayed by the program when practical.
- When distributing binaries, either the source code or an offer to provide source for copying costs only must be provided.

The GNU license clearly states that GPL programs can be included with other programs as parts of larger aggregate collections of programs as long as the GPL programs and the other programs are independent of each other. This allows the GNU programs to be included in Linux distributions that also include non-GPL-compatible programs, for example.

The Free Software Foundation also provides a FAQ document about the GPL to clarify questions about the license. This FAQ can be found at the following URL:

```
http://www.gnu.org/licenses/gpl-faq.html
```

In these FAQs, the FSF makes it clear that linking a program to a GPL library constitutes a derivative work that must also be covered under GPL. In general, if two programs are tightly coupled and share the same process space, and one of the programs is covered under the GPL, then they both must be covered under the GPL. However, if the two programs are loosely coupled and communicate using pipes, sockets, or other means of interprocess communications, they do not both have to be under the GPL. Interpreted programs and scripts that require an interpreter that is covered by the GPL do not themselves have to be covered under GPL.

Another point made clear in the FAQ documents is that the copyright holder is free to release a software product under several licenses. This allows a company, organization, or individual to release a free and open source version of a product under GPL but also release the same product under a different license to those willing to pay not to be under the GPL restrictions. The MySQL database is an example of software that uses this multiple license model.

GNU Lesser General Public License (LGPL)

As stated in the previous section, programs that are linked to GPL libraries are required to also be covered by the GPL. This became problematic for people who wanted to use the GNU compilers to move software to Linux. They could not use the GNU compiler unless the software itself was open source. This could have had a profound effect limiting the use of GNU tools. Instead the FSF created the Lesser GPL (also called the Library GPL) known as the LGPL. This variant has provisions to allow programs to be linked with libraries and redistributed under a separate license. The library itself still needs to be made available as source code under the LGPL, but the combined executable and the library do not.

Mozilla Public License (MPL)

The Mozilla license was created to support the open source release of the Netscape browser. Other companies have used variations on the Mozilla Public License as a model license when they have wanted to release their products as free software. The primary features of the Mozilla license are:

- The MPL allows the use, modification, and redistribution of code.
- The MPL requires that modifications to source code also fall under the same license.
- The MPL requires that source code be documented and made available.
- The MPL allows sections of a program to be covered under multiple licenses.

The FSF recognizes the Mozilla license as a free license but also states that the Mozilla license is not compatible with GPL—with one exception. The latest version of the Mozilla license provides a clause that allows different portions of a software product to be covered under multiple different licenses. This can allow products to be partially Mozilla and partially licensed with a GPL-compatible license.

Open Source Communities

Another common question about open source software is "Where does it come from?" There is often a subtle hint of "Can I trust it?" in that question. The answer to this last question is yes, you can trust open source software, because it is open and free for examination. The answer to the first question is that open source software comes from people, organizations, and companies that invest their time and money in developing these projects. Some projects are sponsored by large reliable companies such as IBM (Eclipse), Sun (NetBeans), SAP (SAP DB), and Netscape Communications (Mozilla). Others are developed by individuals (McKoi comes to mind). Still others are parts of larger open source communities (Apache's Jakarta for example).

Open source software projects have infrastructure needs just as any other project would. Many of these efforts are very large and complex and require coordination among many developers. The open source community provides this infrastructure for the developers. Once established, many projects can often share the same infrastructure, forming a community. These open source communities often serve as incubators for new ideas.

In the following sections, we look at some of the open source communities that have formed over the years to develop and support open source Java development. These communities represent good places to start when searching for open source software. Mostly, our focus will be on projects and communities with a focus on open source Java software.

Apache

The Apache project was originally created to support the development of the Apache Web server. Apache got its name from the fact that it started as a series of patches to the original Cern Web server. The name was a play on words on "a patchy" Web server. Apache is a proven product and has been the most widely used Web server on the Internet for several years, with more sites running Apache than all other Web servers combined. Apache has also become an umbrella for a large number of open source projects that share the Apache license. Often these projects are technology leaders, proving technologies before they are standardized. The Apache project has had a large impact on the open source Java community, initially with the jServ servlet container and now through the Jakarta project. There is more than just one major project under the Apache umbrella that has an impact on Java; we discuss these in the following sections. The main Apache Web page is found at the following site:

```
http://www.apache.org
```

Jakarta

Jakarta is a primary project within the Apache group. Jakarta is focused on developing open source Java code. The primary page for the Jakarta project can be found at:

```
http://jakarta.apache.org
```

Table 1.1 below provides a brief overview of just some of the many subprojects hosted under Jakarta. Some of these subprojects have further subprojects below them. Currently Jakarta represents the largest coordinated community of open source Java projects.

XML

Not all of the open source Java software from the Apache project can be found under Jakarta. A number of Java projects are found under the XML Apache banner at `http://xml.apache.org`. These projects are all of course related to processing XML files. These projects include those shown following in Table 1.2.

Table 1.1 Jakarta Subproject Summary

SUBPROJECT	WHAT IT IS
Commons	Commons is best described as a large collection of small Java utilities. Commons provides a place for those small little classes that are reused time and time again within other Jakarta projects.
James	An email, news, and messaging server written entirely in Java.
Jetspeed	A Java-based Web portal.
Jmeter	A Java-based load-testing tool. Jmeter is covered in Chapter 4.
Log4J	A standard logging library for Java programs.
Struts	A Web application framework used to ease the creation of Web applications. Struts is covered in more detail in Chapter 7.
Taglibs	The taglibs set of projects provides standardized groups of tag libraries for use in developing JavaServer Pages.
Tomcat	An open source Java servlet container and Web application server. Tomcat is covered in more detail in Chapter 5.
Turbine	An alternative Web application framework to Struts. The Jetspeed portal is based on Turbine. Turbine is covered briefly in Chapter 7.
Velocity	A macro-expansion program. Velocity replaces special tags within a document with values that are generated by a Java program. Velocity is covered in Chapter 6.

Table 1.2 XML Apache Projects

SUBPROJECT	WHAT IT IS
Xerces	A Java-based XML parser
Xalan	A Java-based XML style sheet processor
Cocoon	An XML-based Web publishing system written in Java

Web Services

Apache provides a number of projects designed to support Web services. All of the current projects are Java based. These projects can be found at `http://ws.apache.org` and are listed in Table 1.3.

Table 1.3 Web Services Apache Projects

SUBPROJECT	WHAT IT IS
Axis	SOAP-based Web Services for Java.
XML-RPC	XML-RPC-based Web Services for Java.
WSIF	The Web Services Invocation Framework (WSIF) provides a mechanism for creating clients for distributed-processing services by using Web Service Definition Language (WSDL) files.

Ant

Ant is the tool that is most commonly used to build large Java projects. Ant used to be a part of the Jakarta project, but has been promoted to a primary project below Apache. Ant is discussed in more detail in chapter 4 and can be found at: `http://ant.apache.org`.

ExoLab

ExoLab is an informal group focused on developing open source enterprise software projects. ExoLab is probably best known for Castor, a tool that enables XML document binding to Java classes. ExoLab is also responsible for projects such as OpenEJB, OpenJMS, OpenORB, and Tyrex, and contributes to external open source projects such as Tomcat, James, Xalan, and Xerces, which are all a part of the Jakarta project.

The ExoLab Group licenses the code it develops under the ExoLab Public License, which is similar to the Apache License. ExoLab can be found at `http://www.exolab.org`.

GNU

Although the GNU project is not well known for its Java contributions, the GNU project does boast quite a number of ongoing Java efforts. These are mainly focused around providing a pure open source implementation of Java. Many of the Java projects under GNU are hindered by the fact that the Java compiler and JVM are not open source. However, there are still several interesting GNU projects going on, using Java as the core language. The GNU Compiler for Java (GCJ), for example, has the ability to compile Java source code into either Java byte code or machine-native code. The following URL provides a list of Java programs that are linked through the GNU site and are distributed under the GPL.

```
http://www.gnu.org/software/java/java-software.html
```

ObjectWeb

ObjectWeb is an open source community created by a consortium of companies. They have the stated goal of creating and developing Open Source middleware. Two of the more interesting projects are described in Table 1.4. ObjectWeb can be found at `http://www.objectweb.org`. ObjectWeb is now also the hosting organization for the Enhydra project.

Enhydra

Enhydra was one of the first Java-based Web application servers. Enhydra was sponsored by a company known as Lutris that used open source as a means of creating demand for their products. Lutris failed sometime during the dot-com crash, and now the ObjectWeb consortium hosts the Enhydra projects. However the ObjectWeb consortium still host Enhydra at `http://www.enhydra.org`.

Over the years, the Enhydra project has added a number of interesting subprojects. The community now supports a set of projects designed to provide application server capabilities and Web services to Java 2 Micro Edition (J2ME) platforms. Barracuda is a well-regarded Web application framework hosted on Enhydra. XMLC, which provides a clean separation between Web page design and the generation of dynamic content, is also a project worth looking into. Using XMLC a Web page can be mocked up by a Web designer, and the dynamic content can be generated by a servlet. XMLC marries the servlet to the page mockup, replacing the fake values with the real values as the page is served through the application server. This technology can be used to reduce the cycle time on projects using a "visual design, code, and then integrate" cycle.

SourceForge.net

SourceForge.net serves as an open source project incubator, offering free hosting for projects with licenses accepted as open source by the OSI. SourceForge.net is the largest open source community in existence. SourceForge.net currently claims over 56,000 separate projects, over 7,900 of which are Java based. SourceForge.net is owned by OSDN, which in turn is a wholly owned subsidiary of VA Software Corporation. VA Software is known for VA Linux. SourceForge.net can be accessed at (surprise) `http://www.sourceforge.net`.

Table 1.4 ObjectWeb Projects

PROJECT	WHAT IT IS
JonAS	An open source Enterprise Java Bean container
JORAM	An open source implementation of the Java Messaging Service

What makes SourceForge.net so popular? The site's popularity is due to a combination of things. SourceForge.net offers many things that an open source project needs to get started and succeed, and it offers them to open source developers for free. Smaller projects that do not have any funding can get started on SourceForge.net with no startup costs. For example, SourceForge.net provides free Web site hosting for SourceForge.net projects. As a part of this free hosting service, the site offers 100 MB of storage per project. This storage can be used for project-oriented Web content, project files, and documentation. This hosting service also supports CGI and PHP scripting.

SourceForge.net also eases user access to hosted projects by offering a standardized summary page for each project that provides links to the major areas within a project. An example of a summary page is shown in Figure 1.1.

The summary also provides a brief summary of project status and current project issues.

Projects on SourceForge.net are provided with database access to a MySQL database. The database can be used by the project Web site or for developing and testing the project itself. Developers on active SourceForge.net projects also have access to shell accounts through SSH. Tools are available to allow compiling projects directly on the SourceForge.net machines. This means that binaries can be provided for machines and operating systems that normally would not be available to smaller projects.

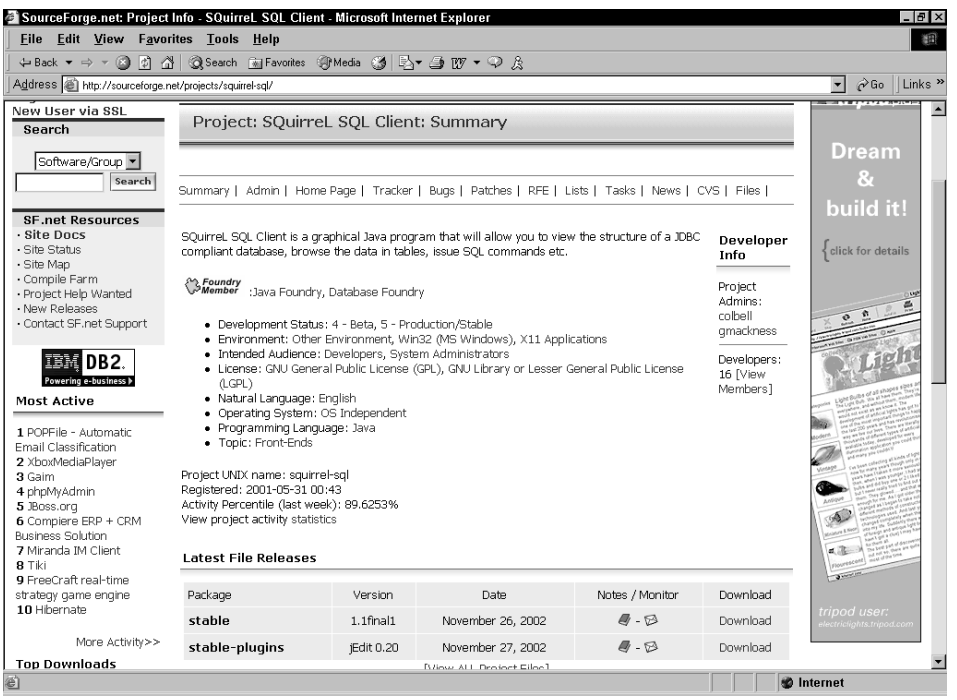

Figure 1.1 Source Forge Project summary page for Squirrel.

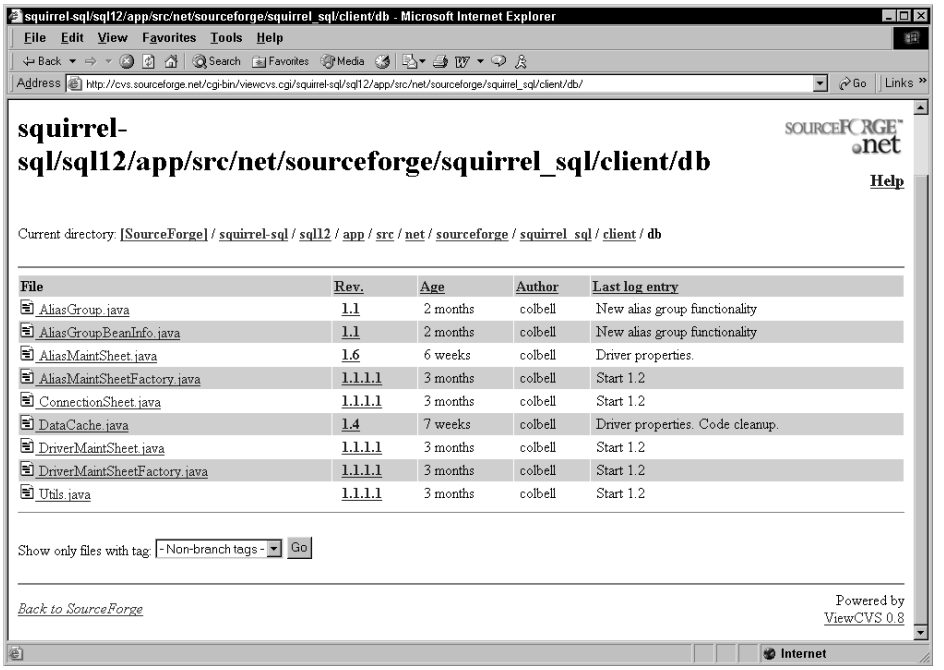

Figure 1.2 Source Forge CVS repository browser.

Each project is provided with a bug-tracking system. The bug tracker allows bugs to be reported by users and tracked by the project team. As progress is made on addressing bugs, the status can be reported back, providing feedback to users, who are able check on the status of the reported bug.

Collaboration is a necessary part of the product mix for any open source community. SourceForge.net provides mailing list support and forum discussion software for each project. The mail list has spam-protection features and is automatically archived providing convenient access to previous list traffic.

SourceForge.net has not forgotten the need for configuration management. It provides each project with a CVS repository. The CVS repositories can be accessed using standard CVS clients. SourceForge.net also provides a Web-based repository browser (shown in Figure 1.2) that allows you to navigate the repository source code tree and view or download individual files as needed.

Summary

In this chapter, we examined the definition of open source software and explored the differences between various open source licenses. These open source licenses will be important to us as we select components to build an enterprise platform. Most of the components we will use come from the open source communities that we briefly explored in this chapter.

In Chapter 2, we will establish the architecture of the platform that we will be creating, and we will identify the components that we will need to create it. In Chapter 3, we will look at the decision-making process for selecting the right components and take a look at some of the factors involved in using open source components in an enterprise setting.