Issue 10/2010
May

http://www.oss-watch.ac.uk/newsletters/may2010.pdf

OSS Watch

Supporting open source in education and research

http://www.oss-watch.ac.uk

IN THIS ISSUE:

Online newsletter available at
http://www.oss-watch.ac.uk/newsletters/may2010.pdf

This month we are focussing on the nuts and bolts of 'doing'. It is often said that the best way to learn about open development is to 'just do it' so we wanted to focus very specifically on the practicalities facing open source projects. To this end Sander van der Waal gives us a whistle stop tour of SourceForge which we hope will be useful for anyone who is considering hosting their project on this well-known hosting site. Sander also has some more practical advice to pass on in his blog piece on how to make your open source software releases more secure whilst Steve Lee tells us about how you can use version control to manage Intellectual Property in an open source project.

We welcome your ideas about topics that you'd like to know more about so if you have ever been stuck for practical information on some aspect of running or contributing to an open source project then please do let us know at info@osswatch.ac.uk.

Elena Blanco, Content Editor, OSS Watch info@oss-watch.ac.uk

News from OSS Watch

SugarCRM moves to AGPLv3

Starting with version 6.0.0, the Sugar Community Edition will now be licensed under the GNU Affero General Public License version 3. 'We agree with the "hosting as commercial distribution" provision that was added to the GPL to create the APGL and feel it applies well to our intention that the Sugar Community Edition source code is to be shared in all circumstances', Sugar explained in a recent blog post.

http://bit.ly/9QdsVd

Twitter open sources FlockDB

Twitter has open sourced the code that it used to build its database of users and manage their relationships to one another, called FlockDB. This follows the release of Twitter's Gizzard framework, used to perform high volume queries of the FlockDB distributed data store.

http://gigaom.com/2010/04/12/twitter-open-sources-the-home-of-its-social-graph/

IBM uses pledged patents against open source mainframe emulator

In 2005, IBM pledged to not use five hundred patents against open source software. In 2010, two of those patents have appeared in a letter from IBM to TurboHercules SAS. The Hercules project, started eleven years ago, is a mainframe emulator. TurboHercules SAS, a French start-up founded by Roger Bowler who started the Hercules project, wants to offer TurboHercules as a disaster recovery solution for IBM mainframe users and wrote to IBM asking for a way to do that. IBM has turned down the request and has accused TurboHercules of intellectual property infringement.

http://www.h-online.com/open/news/item/IBM-uses-pledged-patents-against-open-source-mainframe-emulator-970674.html

Ricoh joins the Linux Foundation

STAY UP-TO-DATE

OSS Watch news feed

http://www.oss-watch.ac.uk/rss/osswatchnews.rss

The Linux Foundation, the non-profit organisation dedicated to promoting Linux, has announced that digital office solutions specialist Ricoh has become its newest corporate member. Ricoh, best known for its multi-function printing products and digital cameras, says that it hopes to further advance Linux, while developing 'more user-friendly devices and useful services for Linux users.'

http://www.linuxfoundation.org/node/6118

OSHUG, new UK user group for open source hardware

OSHUG, the Open Source Hardware User Group, is a new formed group initially based London, but with membership of mailing lists open to all. Osmosoft, the open source arm of BT, will be hosting the first meeting of OSHUG at their offices in London on 29 April 2010. This first meeting will include presentations by Professor David May and Alan Wood.

http://oshug.org/

Moodle enters Macmillan dictionary

The Macmillan dictionary now contains an entry for Moodle: 'modular object-oriented dynamic learning environment: an open source computer system for creating online courses'

http://www.macmillandictionary.com/dictionary/british/moodle

OSS Watch online
OSS Watch blog
Contact OSS Watch
OSS Watch twitter

OSS Watch is funded by the Joint Information Systems Committee
JISC

OSS Watch online

OSS Watch blog

Contact OSS Watch

OSS Watch twitter

OSS Watch is funded by the
Joint Information
Systems Committee
JISC

Creating a project on SourceForge

Full article can be found at http://www.oss-watch.ac.uk/resources/sourceforge.xml

In the early stages of an open source software project, there is much to do in order to lay strong foundations for the project. Some of the most important decisions relate to how that project will be managed, including where to host the version control system, how to set up the mailing lists, and so on. A project hosting website can help you with many of these decisions by providing these and other basic features, as well as several more advanced features, that will help you to manage your project efficiently. Many hosting websites offer their services free of charge to software projects that are released under an open source licence.

SourceForge is one of the best-known hosting sites for free and open source software development projects. The number of registered users of SourceForge passed the 2 million mark in February 2009, a date that also saw more than 230,000 software projects hosted on SourceForge, making this hosting site one of the largest collections of open source tools and applications available online.

> **“Some of the most important decisions relate to how that project will be managed, including where to host the version control system, how to set up the mailing lists, and so on”**

SourceForge offers the basic hosting services that any open source software project needs, such as mailing lists, a version control system and an issue tracker. It makes the process fairly easy by enabling some of its services by default. SourceForge also provides many other services, ranging from web analytics to code review and even a service for URL shortening - an enormous array that can be confusing to new users. This guide discusses some of the most commonly used services and is intended to help you to get started with creating a project on SourceForge.

1. Hosting services overview

Some of the hosting services SourceForge offers are provided by means of so-called hosted applications. These are widely used third-party open source applications which SourceForge offers to its users as a service via its website. For you as a user, there are a couple of benefits to using these hosted applications. Firstly, you will be able to use the full functionality of such an application without having to manage an installation of it yourself. This means that upgrades and security updates are arranged automatically. Also, you can use all of the applications via your SourceForge account, so you don't have to worry about managing separate logins for every application. Below is an overview of all the third-party applications that SourceForge offers, grouped according to their purpose. You can find more information on SourceForge itself.

Function	Application
Blogging	WordPress
Bug tracking	MantisBT, Trac
CMS	phpWebSite
Forum	phpBB
Guestbook	AN Guestbook
Idea brainstorming	IdeaTorrent
Image gallery	Gallery
Microblogging	Laconica
Project management	!dotProject
Surveys	LimeSurvey
Task management	TaskFreak!
URL shortening	sfurl
Version control management	Subversion, Git, Bazaar, Mercurial
Web analytics	Piwik
Web-based Code review	Codestriker
Wiki	MediaWiki, Trac

All of these applications can be activated on the website via an opt-in procedure, making them fairly easy to use. SourceForge offers many more services besides these hosted applications.

In the next few sections we will focus on the most commonly used services, explaining how to get started on SourceForge by creating a project and setting up some basic services.

2. Creating the project

It is quite easy to create a new project on the SourceForge website. This is done by clicking on the menu item *[Create Project]*. When you click on it, you will be asked to log in. You therefore have to create an account with SourceForge first. Alternatively, you can use an OpenId account.

After login, if you have used the *Create Project* link, you will be redirected to a page where you are required to provide the basic details of your new project.

3. Hosted features settings

After you have created your project, SourceForge automatically enables a couple of their services for you. These and all the other services can be managed from the menu item *[Project Admin]* > *[Feature Settings]*.

We advise you to go through this list to check whether you will be using all of these services and to disable those that you will not be using. This will make the layout of your project on SourceForge cleaner, and your visitors will not be distracted by features that are not in use.

Article continues at http://www.oss-watch.ac.uk/resources/sourceforge.xml

Using version control to manage Intellectual Property

Published by Steve Lee on April 23, 2010

Intellectual Property (IP) management is one of the least glamorous activities required when running a software project. And yet it may just provide critical evidence for a quick resolution when a project is forced to defend itself against an IP dispute. A recent example of an attack on a open source project can be seen in the high profile JMRI.org defence against fraudulent claims of patent violations. Proper IP management not only provides peace of mind for the core project team, it also ensures contributors are not individually liable for costs. IP management is particularly important in healthy open source projects as they may receive contributions of uncertain copyright status from a wide range of people of varying and possibly unknown background.

Fortunately, while IP management may seem daunting, not to mention complex, in actual fact a large chunk of the requirements are met almost for free as a side affect of using common development tools. When used wisely, version control systems provide the auditing required to keep track of IP, especially copyright. This is indeed fortunate when you consider that a single contribution may touch a very significant number of the files or other constituent components of a large mature project.

When contributions are accepted into the project code base there is the possibility that some of the code was in fact not usable for legal reasons. Such contamination may come from IP violations such as the process being covered by an enforceable software patent (in the USA), or the contributor being neither the copyright holder nor having been granted appropriate rights. Even if, as recommended, contributors are required to sign a Contributor Licence Agreement to assign ownership to the project itself, there is still a need to track individual contributions in case of dispute.

So a project needs to track who contributed each individual bit of code as it is accepted into the core code base. As a brief aside, the projects governance model will describe who can commit into the core and under what circumstances. The other requirement for IP management is the ability to make various queries about who made a change and when. Any decent version control worth its salt will provide the basic facilities which when combined with simple processes provide basic IP management.

Version control tools by their nature keep track of changes and allow queries to be performed. A project need only ensure that every change is clearly marked with the contributor's identity, bearing in mind that the committer may not be the same person as the contributor. This also implies that commits should be carefully managed so as not to mingle changes from different authors; but that is bad practice anyway for basic auditing purposes. A simple approach is to insist that all modifications include the email address of author. Some tools such as the distributed version control system git make it possible to get such owner stamps to appear automatically in the patches made and submitted as a contribution.

For auditing purposes it may be useful to generate a list of dates of change per contributor and merge that with a list of IP agreements. If any specific part of the code is disputed then the facility to see who made a change and when is useful (often call the 'blame' feature). If a particular change is in dispute it is easy to find out the extent and which files are affected. Finally most version control systems allow notifications to be generated on commits which can be useful for notifying those responsible for checking IP.

Our article What is version control? Why is it important for due diligence provides more detail on using version control.

http://osswatch.jiscinvolve.org/2010/04/23/using-version-control-to-manage-intellectual-property/

Making your open source software releases more secure

Published by Steve Lee on February 18, 2010

When you're writing code in an open source software project, you are generally using some version control system (you should!). Hosting websites like SourceForge and Google Code usually provide one free of charge, eg. a Subversion repository. All developers or other people interested in the source code use a client application to download the code and synchronize with the repository on the server. Users can download the source code using http or https.

Making your code available for external parties by using a version control system is a very good idea. But if you want to attract more kinds of users than just developers, you should also periodically create a release of your source code and make these separately available for download. Less or non-technical users will then find it easier to use your software and get involved in the project more easily if they don't have to build the executables themselves.

However, if you release executables through your own (or a hosting) website, this is not entirely risk-free. If someone tempers with your binaries, eg. by adding malware to the code, your users may be downloading malicious code to their computers by using your software, and of course you would want to prevent that. Two ways of doing this are to hash the download and make the checksum available as a separate file, or by signing the release with a PGP signature.

Adding a checksum to your download

Several algorithms have been developed to generate a hash checksum for a file, eg. MD5 or the SHA family (SHA1, SHA128, SHA512). The purpose of creating this message digest is to assure that the file you downloaded is exactly the same file that was offered, so no one byte can be different. This does not only help against infringement from outsiders, but also helps you detecting technical errors in the transmission of the file. An easy, simple-to-use tool to generate hashes is the open source application GNU Privacy Guard. It supports MD5 as well as several SHA algorithms. In general you should use the algorithm with the longest hash key, as they are more secure. With brute force it is possible to break all hash keys, but a short key like from MD5 is much easier to crack than one like SHA512. The process is fairly simple. You create the checksum file for the generated download and upload them both to your website. The user interested in the download also downloads the checksum and uses a similar tool as GNU PG to check if the checksum is correct.

Signing your release with a PGP signature

Another way to protect your released binaries is to sign them with a PGP signature. You can use the same tool GNU Privacy Guard for that. When you start using it, you first need to generate a public/private key combination. You will have to publish the public key in a file, commonly this is one KEYS file for the whole project that contains the public keys of all relevant developers. Next, you can sign a release binary using your own key, which will result in a signature file, which is quite similar to a hash key file. Now anybody that can download the binary, your public key and the signature file, can check whether the signature matches with the public key and the binary. This will ensure that the file that has been downloaded is the same file that has been created by you, the release manager. PGP signatures can provide an extra level of protection by identifying that the public/private key combination that signed the binary indeed is linked to the individual. It works in a decentralised way (as opposed to the PKI method that needs a certifying authority). Once you have created your own keys you can exchange your public key with other users and thereby add them to your web of trust. You do that by signing each other's keys, but you should really do this in person to make sure you know whom you are dealing with, e.g. at a conference.

Finally, if you'd like to know more the Apache page on signing is an excellent resource. For more technical details about signing Henk Penning's page is very informative.

Then, as a really final note, I should add that these mechanisms of securing your releases are not error-free and don't fully guarantee that people with malicious intentions can do harm, because they usually always can. However, the described mechanisms do make your releases safer than not using any security and it's therefore a good idea to use them.

http://osswatch.jiscinvolve.org/2010/04/21/making-you-open-source-software-releases-more-secure/

Events

STAY UP-TO-DATE

OSS Watch events feed

http://www.oss-watch.ac.uk/rss/events.rss

June
26

BarCamp Oxford, 26 June 2010

OSS Watch has teamed up with Torchbox to run a BarCamp at the Oxford University Club in Oxford in 26 June 2010. In keeping with the concept of a BarCamp this is an informal, geeky event where delegates can set their own agenda for discussions. This BarCamp follows on from the TransferSummit taking place in Oxford over the previous few days so we hope to see some of the conference delegates staying on for the BarCamp but you don't need to have been to the conference to join in the fun. Why not sign up today!

http://www.barcamp.org/BarCampOxford

June
24-25

TransferSummit, Oxford, 24-25 June 2010

This summer, OSS Watch is sponsoring a major open source conference, aiming to connect academia with open source businesses. The main conference will be held over two days, 24-25 June 2010, at Keble College Oxford. Covering topics within both academia and business, the event will try to identify areas of activity of mutual interest, looking at how the two sectors can engage with each other. Registration is now open.

http://www.transfersummit.com/

Frequently Asked Questions

Q Is the model of open source software development useful for academic research?

A In general, yes. There are many examples of open source software that has been developed by and for researchers, e.g. TexGen. An open development culture can also be very beneficial in a collaborative research environment.

Q What is a governance model and how do I design one?

A A governance model is a public document that describes how a project is managed. In particular it describes the structure of the team including individual roles and clearly describes how others may contribute to a project. It also outlines the processes that are followed when performing project activities.

While there is potential for an infinite variety of governance models they tend to fall somewhere on a scale between the two commonly recognised extremes known as the 'meritocratic' and the 'benevolent dictator' models. The difference between these two models is, in fact, not so great and mostly concerns the the mechanism for resolving conflict in the decision making process.

For more answers to your questions visit: http://www.oss-watch.ac.uk/about/faq.xml

- OSS Watch online
- OSS Watch blog
- Contact OSS Watch
- OSS Watch twitter

OSS Watch is funded by the
Joint Information
Systems Committee
JISC