

Survey on Version Control

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. If you are a graphic or web designer and want to keep every version of an image or layout (which you would most certainly want to), a Version Control System (VCS) is a very wise thing to use. It allows you to revert selected files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more. Using a VCS also generally means that if there is any mistake or lose files, you can easily recover. In addition, you get all this for very little overhead.

Different types of version control :

Git

Git is considered to be a super-fast and efficient, cross platform version control tool. First developed by the creator of Linux kernel, Linus Torvalds, Git has begun to take the community for web development and system administration by storm, offering a largely different form of control. Here, there is no singular centralized code base that the code can be pulled from, and different branches are responsible for hosting different areas of the code. Other version control systems, such as CVS and SVN, use a centralized control, so that only one master copy of software is used. It provides strong support for non-linear development. It is also compatible with existing systems and protocols like HTTP, FTP, ssh. For security, cryptographic authentication of history is done. It provides periodic explicit object packing.

As a fast and efficient system, many system administrators and open-source projects use Git to power their repositories. Code changes can be very easily and clearly tracked, maintainable and robust. Offers an amazing command line utility known as git bash. However, it is worth noting that Git is not as easy to learn as SVN or CVS is, which means that beginners may need to steer clear if they're not willing to invest time to learn the tool.

CVS

CVS is a "Source Control" or "Revision Control" tool designed to keep track of source changes made by groups of developers working on the same files, allowing them to stay in sync with each other as each individual chooses.

CVS uses a centralized, client-server model for source control. This means that the authoritative copy of a code repository is stored on a server, and developers connect to that server and "check out" code to work on. Client computers check out a full, working copy of the code from the server, and then check-in their changes. CVS automatically handles version numbering on individual files, based on changes made at check-in. It provides an excellent cross-platform support, robust and fully-featured command-line client permits powerful scripting, helpful support from vast CVS community, allows good web browsing of the source code repository. Therefore it suits the collaborative nature of the open-source world splendidly.

Unlike some other centralized version control systems, CVS does not lock files when they are checked out. Multiple developers can work on the same files at the same time. However, when you check-in your changes, CVS will only accept changes if they were made to the most recent version of a file. This requires developers to pull in changes from the central repository on a regular basis, and encourages frequent check-ins. Client tools usually automate this task, pulling in changes from the central repo on schedule, and merging non-conflicting changes automatically. However, it's main disadvantage lies in poor support for distributed source control, no support for signed revisions and merge tracking.

SVN

Apache Subversion (often abbreviated SVN, after its command name *svn*) is a software versioning and revision control system distributed as open source under the Apache License. SVN is generally the version control system that has the widest adoption. Most forms of open-source projects will use Subversion because many other large products such as Ruby, Python Apache, and more use it too. Google Code even uses SVN as a way of exclusively distributing code. It provides merge tracking, full MIME support, path-based authorization, file locking, standalone server operation, free-form versioned metadata, space efficient storage, directories are versioned, copying, deleting, moving and renaming operations are also versioned. It also supports atomic commits, integrates well with Windows, leading IDE and Agile tools. However, it does not store the modification time of files, does not deal well with filename normalization, neither does it support signed

revisions. Because it is so popular, many different clients for Subversion are available. If you use Windows, then Tortoise SVN may be a great browser for editing, viewing and modifying Subversion code bases. If you're using a MAC, however, then Versions could be your ideal client.

Mercurial

This is yet another form of version control system, similar to Git. It was designed initially as a source for larger development programs, often outside of the scope of most system admins, independent web developers and designers. However, this doesn't mean that smaller teams and individuals can't use it. Mercurial is a very fast and efficient application. The creators designed the software with performance as the core feature.

Aside from being very scalable, and incredibly fast, Mercurial is a far simpler system to use than things such as Git, which one of the reasons why certain system admins and developers use it. There aren't quite many things to learn, and the functions are less complicated, and more comparable to other CVS systems. Mercurial also comes alongside a web-interface and various extensive documentation that can help you to understand it better

Bazaar

Similar to Git and Mercurial, Bazaar is distributed version control system, which also provides a great, friendly user experience. Bazaar is unique that it can be deployed either with a central code base or as a distributed code base. It is the most versatile version control system that supports various different forms of workflow, from centralized to decentralized, and with a number of different variations acknowledged throughout. . One of the greatest features of Bazaar is that you can access a very detailed level of control in its setup. Bazaar can be used to fit in with almost any scenario and this is incredibly useful for most projects and admins because it is so easy to adapt and deal with. It can also be easily embedded into projects that already exist. At the same time, Bazaar boasts a large community that helps with the maintenance of third-party tools and plugins.

Reason for choosing GIT

- Git has the advantage that it's MUCH better suited if some developers are not always connected to the master repository.
- It's much faster than SVN. And from what I hear, branching and merging support is a lot better.
- Git is perfectly suited for Open Source projects.