Git and Version Control

- Subversion
- Revision Control
- Revision Control Systems Compared
- Installing Subversion on Windows
- GIT info
- What are some document management services/document version control applications out there?
- svn: Working copy '<filename>' is missing or not locked

Subversion

What is Subversion?

Subversion is a version control system or SCM (software configuration management) tool. It is open-source and provides a feature set superior to that of CVS and Visual Source Safe. More information can be found at subversion.tigris.org.

Tools for use with Subversion

General tools

- The most popular and well-maintained Windows Subversion tool is TortoiseSVN. Tortoise is a Windows shell extension that integrates Subversion commands into Windows Explorer.
- NaughtySVN is a shell extension for the Gnome project's Nautilus file manager.
- SCPlugin is a shell extension for Mac OSX, but as of this writing it is not up to date.
- RapidSVN is a cross-platform standalone Subversion GUI.
- SvnX is an OSX based tool similar to RapidSVN.
- ZigVersion – standalone client like svnX but some think it's quite a bit easier to use.
- Subversion itself is cross-platform and features a very well-documented command line interface.

IDE integration

- Subclipse is a Subversion plugin for Eclipse.
- AnkhSVN is a plugin for Microsoft Visual Studio, but it is no longer in active development. TortoiseSVN is the recommended tool for Windows/Microsoft developers.
- Apple has some information for using Subversion with XCode

The Subversion Book / Documentation

- The entire O'Reilly book on Subversion is freely available online.
- Bernie Zimmerman has written a very simple tutorial about Installing Subversion on Fedora Core 3
- Software Carpentry – Version Control with Subversion – Screencast How To Video
- https://kb.ucla.edu/link/514
- Installing Subversion on Windows
- Subversion Best Practices
- Making the Most of Commit Hooks with Subversion – Linux Magazine, Oct. 9, 2008 – Registration Required

Revision Control

What is Revision Control?

Revision control (also called version control or source control) is a method of tracking changes between various versions of a digital document. This is typically used for software code, but can be used for any type of digital document. For example, this knowlege base article tracks changes between versions, as do Wiki sites.

Revision control for software is a vital part of Software Configuration Management or SCM. Bug tracking can also be considered a part of SCM.

Why should I (a programmer) use Revision Control?

The basic reasons are:

- Allows undoing and tracking of changes
- Makes it much easier to manage release and development versions of software
- Greatly facilitates more than one person working on a piece of software
- Promotes good programming practices

If you're not convinced, read this article: Why do You Need A Version Control System? .

There is ***no reason*** not to be using Source Control, even if you are a single developer.

Links / Documents

- Wikipedia's entry on Revision Control is a good introduction to the concept and has a very useful glossary.
- Eric Sink has written an excellent and in-depth series of articles on source control . The articles focus on general version control concepts rather than specific tools.
- CVS Best Practices apply to CVS as well as other Revision control systems.

- This knowledge base article features an article listing and comparing some popular revision control systems.

Revision Control Systems Compared

Comparison charts

There is an in-depth Version Control Systems Comparison available that covers many important features you might be looking for.

Popular Systems and their Pros/Cons

This list is by no means complete. For more in-depth information, refer to the link above.

Visual SourceSafe

SourceSafe is Microsoft's own source control tool. If you do all of your development in Visual Studio, this is the easiest (but not necessarily best) tool for you.

Pros:

- Excellent integration with Microsoft Visual Studio (Ease of Deployment)
- Many microsoft developers already have access to it
- Relatively easy to use

Cons:

- Too simple for many users
- Commits are not atomic
- No support for changesets
- Not portable
- Proprietary and non-free

CVS

CVS is a very widely-used system. You should only use CVS if your environment has limitations that keep you from using SubVersion.

Pros:

- Portable
- Excellent tool support
- Open-source / free
- More features than SourceSafe

Cons:

- Dated – lacking many modern features
- No atomic commits

Subversion

Subversion is a newer open-source tool designed to replace CVS. It is superior to CVS in almost every way and is the best all-around open-source tool of its kind. There is an article in this knowledge base with more information about Subversion.

Pros:

- All the pros of CVS plus
- Atomic commits
- High performance
- Vast array of configuration options
- Handles binary files efficiently as well

Cons:

- Can be complex to setup depending on the configuration chosen
- No good Visual Studio integration yet

Perforce

Perforce is an easy to deploy and very feature-packed proprietary solution. It is used by Google. A free version is available, but it only supports a maximum of 2 users.

Pros:

- Excellent feature-set
- Atomic commits
- Very good IDE integration (including Visual Studio)
- High performance
- Handles binary files efficiently as well

Cons:

- Expensive

Others

One of the tools listed above will most likely be right for you, but they are not the only tools out there. There is no single best choice for a source control system. It is important to use the tool that is best for you. BitKeeper was used to manage the Linux kernel, Git (designed and developed by Linus Torvalds) is currently used to manage the Linux kernel, and ClearCase by Rational is also popular in many large companies.

Distributed Version Control Systems

- http://en.wikipedia.org/wiki/Distributed_Version_Control_System
- Why distributed version control
- Choosing a Distributed Version Control System

Thanks to Jose Hales-Garcia for his post on this on OSXForum .

Future Expansion

Someone with more information about Visual Studio Team System should add something about that.

Installing Subversion on Windows

- Download and install the Windows binary.
- Download and install svnservice.
- Because we will run Subversion as a service, *C:\Program Files\Subversion\bin* must be part of the **PATH** environment variable. The installation should have done this but double-check this.
- Create a repository:
  svnadmin create "C:\IT\Subversion Repository"
- Make sure file permissions on this folder are locked down. Edit the conf/svnserver.conf file:
  [general]
  anon-access = read
  auth-access = write
  password-db = passwd
  [users]
  ${username1} = ${password1}
  ${username2} = ${password2}
  ${username3} = ${password3}
- Run *SVNService Administration* and fill in location of Subversion bin folder and the repository you created above. Notice the port (3690). *Listen Host:* should be set to the FQDN of the server so that it can be accessed remotely. Start the service. Change firewall settings to allow connections to this port as desired.
- Install Tortoise SVN on a client computer.
- Create a project with the usual subfolders:
  - branches
  - tags
  - trunk

External Links

- http://blogs.vertigosoftware.com/teamsystem/archive/2006/01/16/Setting_up_a_Subversion_Server_under_Windows.aspx
- http://www.stanford.edu/~bsuter/subversion-setup-guide/

GIT info

Please add other helpful links:

http://www.git-tower.com/blog/git-cheat-sheet-detail/

Interactive tutorial about branching – (Learning about git branching)

Mostly for beginners – (List of Git Tutorials from 2011)

Role specific command references for different types of git users- (Everyday git commands from kernel.org)

10 Tips to Push Your Git Skills to the Next Level – June 17, 2014

What are some document management services/document version control applications out there?

Two great options that we recommend are Git and SVN.

Git: http://git-scm.com/
Git is a free and open source distributed version control system designed to handle everything from small projects to very large projects with speed and efficiency. It offers a variety of exclusive features such as cheap local branching, convenient staging areas, and multiple workflows.

SVN: http://subversion.apache.org/
Apache Subversion is a full-featured version control system originally designed to be a better CVS. Subversion has since expanded beyond its original goal of replacing CVS, but its basic model, design, and interface remain heavily influenced by that goal. Even today, Subversion should still feel very familiar to CVS users. For more information on what CVS is, refer to the following link:

http://ximbiot.com/cvs/

svn: Working copy '<filename>' is missing or not locked

Problem: While doing a *svn update*, you get the following message: “svn: Working copy ‘[filename]’ is missing or not locked”

Cause: The directory that the file [filename] is in needs executable (i.e. list) permission on for the user that issues the command.

Solution: Set the execution permission, e.g. chmod +x [dir_that_contains_that_file]