

# Version control with SVN and TortoiseSVN

## Getting started with TortoiseSVN:

### Installing:

1. Go to <http://tortoisesvn.net/downloads> and download the current stable \*.msi file for your architecture, presumably 32bit in this case.
2. Double click the resulting \*.msi file and follow the instructions which should culminate in rebooting your computer.

### Checking out a test repository:

1. Before we can check out the repository, you are going to need to have the SVN tools on lore in your path, so open your preferred terminal emulator and login to lore.cs.purdue.edu.
2. Once you're logged in, add the following line to your .zshrc, .bashrc, .tcshrc, or whichever shell configuration you use.  
for sh, bash, or zsh  
`export PATH=$PATH:/p/subversion/bin`  
for csh or tcsh  
`setenv PATH=$PATH:/p/subversion/bin`
3. Save the file and log out of lore.
4. Create a directory on the desktop called 'test'.
5. Right click on the test directory and select 'SVN Checkout'.
6. Our test repository is at /u/u3/cs490t/Fall2006/svn, so in the url field you should have 'svn+ssh://(username)@lore.cs.purdue.edu/u/u3/cs490t/Fall2006/svn' where (username) is your unix login. Then hit 'Ok' and when asked enter your password.
7. You should now find the LaTeX source for this handout and a directory labeled '.svn' in the test directory.

### **Adding files and checking in new revisions:**

1. Now that you have checked out our test repository create a file named (username).txt where (username) is your unix login, you can place text in it if you like though that is not really necessary.
2. Right click on your new text file, and go to TortoiseSVN->Add... then click ok on the two windows that pop up. At that point you should notice that the file now sports a blue plus sign, signifying that you have marked it to be added on the next commit.
3. Now, since you have added a file you want to commit, but one thing should always be done before checking in a revision, and that's running an update. So, right click on the test directory and select SVN Update.
4. Once the update is complete it should be safe to commit your new version, so right click on test again and select svn commit. That will present you with a commit message window, usually it's good to have descriptive messages for commits so that if you want to come back later and find a change, the comments can make it a much easier job, for today something like "adding (username).txt" should be fine. once you have entered a message click ok, and the file will be added to the repository.

**Not covered in this document:** One topic you will need to deal with is merging code. Whenever you run an update, SVN will attempt to automatically merge your local changes with those committed to the server since you last updated, usually this will work fine and you won't have to do anything extra, but sometimes there will be conflicts, where you and someone else modified the same place in the same file in different ways. When that happens you will need to merge the files manually, doing this with TortoiseSVN is not terribly complicated but is beyond the scope of this document, you can find more information on [tortoisesvn.tigris.org](http://tortoisesvn.tigris.org) and if you have trouble feel free to ask me for help.

**Making it easier:** TortoiseSVN is a relatively simple tool, but sometimes things can be a little annoying, like typing in your password over and over, one way around this is that TortoiseSVN can integrate with Pageant from the putty suite. If you download the PuTTY suite, you can use PuTTYgen to generate a DSA key, then add the public key to your `/.ssh/authorized_keys` file on your laptop, then when you open the key file with pageant, you won't need to

use your password to update or commit your sources (or log into lore with PuTTY for that matter).