Subversion Overview and Architecture

Subversion is a great system for source code management. You can add new code, revert back to old code, revert back to even older code, and propagate changes to everyone working on the project. This will be incredibly useful for your project. If you do not use it, you will be at a significant disadvantage. So use it!

Subversion works in a server/client model. You set up a central 'repository' which all of your clients can access. This makes it perfect for working on several different computers, or working with a group of people. After setting up the repository (sometimes called a repo), project members can set up the initial layout by 'checking out', add files to it by 'adding', make changes to existing work by 'commiting' changes, and get the latest version of code by 'updating'. If you're working with a group, it's the best way to go. You'll be able to stay current with what other people are doing, see the code changes they've made, and give them your changes without e-mail, thumb drives, or anything like that. So let's get started.

Subversion Setup

If you're a Windows user, you will want to get TortoiseSVN. It is a great visual client for Subversion. In fact, stop reading right now and go download that.
If you're a Mac user using Eclipse, you should probably look into Subclipse. I have never used it, but I've heard good things about it. It is an Eclipse plug-in, so it should work well with your IDE. The following instructions are for command line use of subversion, which will not require anything extra, and is also pretty easy to use.

First, pick a person who will use his ENIAC space to house the repository.

This person will use his or her eniac account to set up the repository. You will also need your UNIX group name.

Remember, you are NOT going to house this on your own machine. This is going to live on the ENIAC family of servers. So the following instructions need to be executed after logging into ENIAC.

The person who will be housing the repo(sitory) needs to log on to ENIAC
1) **mkdir <repo_name>** //creates the repo directory
2) **chgrp <group_name>** <repo_directory> //puts the directory into your group
3) **chmod g+s <repo_name>** //this will add all newly created files to the group
4) **svnadmin create <repo_name>** //create the subversion repository
5) **chmod -R g+rw <repo_name>** //gives read and write access to the group

That creates your repository. Now you need to create a directory (on your own machine, or eniac, either is acceptable) structure for your final project. You can add directories later, but it's nice if you create an initial directory structure. I usually put this in a temporary directory with no files.

topdir
-> deformer
-> visualizer
-> geometer

Then go into topdir and import that structure into your repository.

If you use TortoiseSVN, and you created the directory structure on a Windows machine, right click on the directory, go to the TortoiseSVN options, and click import. Here is the URL of your repo:
svn+ssh://<username>@eniac.seas.upenn.edu/home1/<first_letter_of_creator_id>/<creator_id>/<repo_name>

If you use Mac, you can use a visual client, but I'm not sure of what ones are good. You can use the command line pretty easily though. Open up a terminal and navigate to 'topdir'. then type the following:
svn import svn+ssh://username@eniac.seas.upenn.edu/home1/<first_letter_of_creator_id>/<creator_id>/<repo_name> -m "New Import" (that is all one command that just stretched across two lines)

You should see the files importing into your repository. That means it worked.

Now you want to check your files out to make sure it really worked.
Windows: Right-click on the directory, and click **SVN Checkout**. Specify the same URL as before, and give your ENIAC password. Done and done. If you get a privileges error, you did not set up the permissions correctly initially.

Mac/Linux: bring up a terminal window, go to the directory where you would like to check out your source, and type the following:
"**svn co <svn_url>**"
Done and done. If you get a privileges error, you did not set up the permissions correctly initially.

Now you have your initial check out. When you create new files, you need to add them to your repository. Unfortunately, SVN will not automatically do this for you.
Windows: Select all the files you want to add, right click, select **TortoiseSVN->Add**.
Mac/Linux: type **svn add <file_name1> <file_name2> ... <file_namen>** with all of the new files you want to add.

Commiting new changes to the repo is pretty easy.
Windows: Right click on the file or directory you would like to commit, and click **SVN Commit**. You can right click on the top-level directory to commit all changes in all subdirectories
Mac/Linux: type **svn commit <file_name1> <file_name2> ... <file_namen>** or just type **svn commit** in any directory where you would like to commit all files. If you do this at the top level directory, you will commit all files in all directories below.

Updating to the current source is also easy.
Windows: Right click on the topmost directory, and click **SVN Update**.
Mac: At the topmost directory, type **svn update**.

Subversion Tips:

Tip #1: Definitely use Subversion. It will make your integration easier, as everyone will be able to work with code updates as they come in. Students in last year's class said that groups who did not use SVN had lots of problems.
Tip #2: Don't check in broken code unless you absolutely have to and you

warn everyone else. If you really want to backup your progress through Subversion, but your code isn't currently working, you can do that. But make sure you tell everyone that they shouldn't check out your new stuff.
Tip #3: You don't always have to check out the entire project, as long as you divide things up nicely. If the camera person has made some great improvements, but the deformer just broke his section, you can only update the camera code if there's a directory where all of that code is stored. If all of the code is just plunked together in the same place, you're going to have a problem.
Tip #4: As Norm said, Subversion is not a replacement for communication. Make sure you add messages to what you check in so that other people know what changes were made. Also, you can still let people know that you're checking in some new changes and what they were.
Tip #5: Come talk to us sooner rather than later. We can help you set this up. Getting it set up earlier means you can get integrated faster. It also gets you thinking about the project that much sooner.

These are the two links provided by CETS. They do not go over the steps in order, so do not follow it to the letter. But it is a good reference. Also, you can find out just about anything you want to know about Subversion (aka SVN) from Google.

1) http://www.seas.upenn.edu/cets/answers/subversion.html
2) http://www.seas.upenn.edu/cets/answers/setgrp.html