

# Intro to Source Control Management (SCM)

## Overview

You will be looking at the basics of using a source control management (SCM) client (a.k.a version control) using the “Tortoise SVN” windows subversion client. SCM allows multiple developers to be working on the same project and alter the same files, with all changes being merged together and stored in a central location.

1. Sign up for a free account at <http://www.xp-dev.com/>
2. Inform the lecturer of your username on this site so you can be added to the project – if the lecturer is unavailable, continue working on other practical exercises until they become available.
3. Using a subversion client, checkout the ‘Intro2SCM’ project ... the following are instructions for Tortoise SVN on Windows, if you are using a different client or operating system, please see your tutor for help:
  - 3.1. In windows explorer, browse to the folder you want to check out the project to (a new folder containing the project will be created).
  - 3.2. Right-click in the window to bring up the context sensitive menu and select SVN Checkout (or do this from the File menu).
  - 3.3. The repository URL should be entered as <http://svn.xp-dev.com/svn/Intro2SCM/>
  - 3.4. The checkout directory should be the current path in explorer, with ‘Intro2SCM’ at the end. You can change the name at the end of this path (if you want) as this will just be the name of the new folder to be created. E.g. checkout to ‘X:\Uni Work\UWS\Intro2SCM’
  - 3.5. All other setting should be fine, so just click OK
  - 3.6. When asked for your username and password, enter the details from step 1. You should not save authentication if using a public computer. If you do save the authentication, it should only be for this project though.
  - 3.7. You should now have a new folder created in explorer that has an icon overlay (usually a green tick) to show it is part of a subversion repository.
  - 3.8. Inside the folder is a text file, ‘Testing.txt’. Add a line at the end with your name and your comment. If the formatting doesn’t allow enough space for your name before the comment, add some tabs in on the other lines to give you enough space.
  - 3.9. Save the file and return to windows explorer. The icon overlay for the file should have changed to show that the file contains an uncommitted change. This will probably be a red exclamation mark, but depends on the icon set being used.
  - 3.10. Go up one level so you can see the project folder, which should also have a changed icon overlay to indicate that the contents of the folder have been changed and need to be committed to the repository.
  - 3.11. Right-click on the project folder and choose SVN Commit

- 3.12. In the commit window, type in a commit message (this is retained in the version logs so that anyone checking the history can see why this change was made) and then click OK
- 3.13. Again you will be asked for your username and password, enter them as before (unless you saved these details earlier).
- 3.14. If all goes well, your changes will be committed to the repository, and anyone checking out the project now will see the changed file. There may, however be a conflict when you trying committing:
  - 3.14.1. If another user has committed a change between the time that you checked out the project (or updated it) and the time you tried to commit, a conflict occurs. This is to prevent users from accidentally overwriting changes made by other users. Both the file and the containing folder will be marked as conflicting, probably with a yellow triangle and black exclamation mark.
  - 3.14.2. Right-click the project folder and choose SVN Update, again entering your user details. This should take the most recent committed version of the text file and merge your changes with it. You can then open the file to see the updated text, including your changes.
  - 3.14.3. Now that you have the merged spreadsheet, try committing the project again.
4. If all is working fine, why not explore what happens if you add some extra (small) files to the repository, or try out the other SVN features. Don't worry, this is just for testing, so if anything goes horribly wrong then it can be repaired.

Please note that although there is some built-in merging functionality in SVN, there may be irresolvable conflicts if many people edit the same line at the same time and try to commit, so a file diff and merge tool such as Kdiff3 (<http://kdiff3.sourceforge.net/>) would come in handy. If you have any questions or something isn't working right, let your tutor know.