soundsoftware.ac.uk

Version Control

What is it?

Software to help keep track of changes made to files

- Tracks the **history** of your work
- Helps you **collaborate** with others

Keeping track of history

- How do you get back to that working version you had yesterday?
- How do you get from “it’s not working!” to understanding what went wrong?
- How would you repeat the experiments from that journal paper you wrote last year?

Collaborating... with yourself!

You need to run the same software on your laptop at home and the server in the lab:

- How do you get the code onto both?
- How do you *verify* that you have the same code on both?

Collaborating with others

You're working on code or a paper with a colleague...

- How do you find out when they change something?
- How do you merge your changes without getting in a mess?
- How can you find out which of you introduced a bug, and when?

Version control helps with all this

But you do have to work a bit:

- Tell it which files are part of your project
- Tell it when you've changed something
- If two people make *conflicting* changes, one of them has to *resolve* them

Version control terminology

Repository - the complete history of your project: every version of every file

Working copy - the project as you're working on it now

Add a file you want the system to track

Commit to record a change, making a new *revision*

Update to make your working copy reflect a different revision from the repository

Push or *pull* to synchronise copies of a repository

Time to get cracking

Managing work

- When should I start using version control for my project?
- Which files should I track in the repository?
- How often should I commit?
- How often should I push?

Version control systems

Subversion: Centralised

- one server, one repository database
- every commit goes straight to the shared repository

Mercurial, git: Distributed

- every working copy has complete repository in it
- commits are local, then push/pull between computers

Further reading

- Why use version control?
  http://soundsoftware.ac.uk/why-version-control
- A Mercurial tutorial (command-line)
  http://hginit.com
- Bite-sized Mercurial
  http://petevidler.com/series/mercurial/
- EasyMercurial videos
  http://easyhg.org/videos.html
- Software Carpentry version control (Subversion)
  http://software-carpentry.org/4_0/vc/