



Software Engineering Portfolio 5

Student ID: A3094

Student Name: Lim Zhan Qing

Module Lecturer: Dr Nurazlin Bt Zainal Azmi

Version Control

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. For the examples in this book you will use software source code as the files being version controlled, though in reality you can do this with nearly any type of file on a computer.

If you are a graphic or web designer and want to keep every version of an image or layout, a Version Control System is a very wise thing to use. It allows you to revert files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modifies something that might be causing a problem, who introduced an issue and when, and more. Using a VCS also generally means that if you messed up or lose files, you can recover by just pressing a button or command (depending on the platform you are using). In addition, you get all this for very little overhead.

Local Version Control Systems

Many people's version-control method of choice is to copy files into another directory (perhaps a time-stamped directory, if they're clever). This approach is very common because it is also incredibly error prone. It is easy to forget which directory you're in and accidentally write to the wrong file or copy over files you don't mean to copy.

One of the more popular VCS tools was a system called RCS, which is still distributed with many computers today. Even the popular Mac OS operating system includes the RCS command when you install the developer tools. RCS works by keeping patch sets in a special format on disk; it can then re-create what any file looked like at any point in time by adding up all the patches.

Why is VCS important

It is common that a project is handled by many people, and the changes everyone has in mind are different. But in VCS, working on a project as a team cannot get easier as it provides a huge personal space for developers to do whatever they want with the project. More advantages are listed below

- **Version Tracking** – Developers may wish to compare today's version of some software with yesterday's version or last year's version. Since version control systems keep track of every version of software, this becomes a straightforward task. Knowing the what, who, and when of changes will help with comparing the performance of particular versions, working out when bugs were introduced, and so on. Any problems that arose from a change can then be followed up by an examination of who made the change and the reasons they gave for making the change.
- **Coordinating Teams** – Resource development is usually carried out by teams, either co-located or distributed. Version control is central to coordinating teams of contributors. It lets one contributor work on a copy of the resources and then releases their changes back to the common core when ready. Other contributors work on their own copies of the same resources at the same time, unaffected by each other's changes until they choose to merge or commit their changes back to the project. Any conflicts that arise - when two contributors independently change the same part of a resource – are automatically flagged when the changes are merged. Such conflicts can then be managed by the contributors. Typically in open source projects, version control systems allow anyone to read and copy the project resources, but only authenticated users, known as committers, are allowed to update source code in the repository.
- **Due Diligence** – Many activities in business are accompanied by a responsibility to perform 'due diligence' checks. Precisely what these checks entail will depend on the business activity in question, but with regard to intellectual property one important 'due diligence' activity is the tracking of the ownership of its constituent parts. So for example, if someone creates a piece of software and wishes her organization to release it, her organization will almost certainly want to check the provenance of all the code within the software. This process is facilitated by the ability to track who made which changes to the code, and when they were made. A version control system enables a list of contributors to be compiled and the dates of their contributions to be ascertained. Such a list can be easily cross-checked with a list of IP contracts.

Function of Version Control

Basic Setup

Repository – The database storing files. Usually a website that stores the file that you created

Server – The computer used to store the repository.

Client – The computer connecting to the repository. This is a situation where two or more individual is working on one repository.

Working Set – Local directory of files, where you or another collaborator make changes.

Trunk/Main – The primary location for code in the repository. It is like a code as a family tree – the trunk is the main line.

Basic Actions

Add – Insert a file into the repo.

Revision – Check the current version of a file.

Head – The latest version of a file in the repository.

Check Out – Download file from the repository.

Check in – Upload a file to the repository. The file gets a new revision number, and other collaborators can download the latest one.

Check-in Message – A short message describing what was changed (usually entered with the commit command).

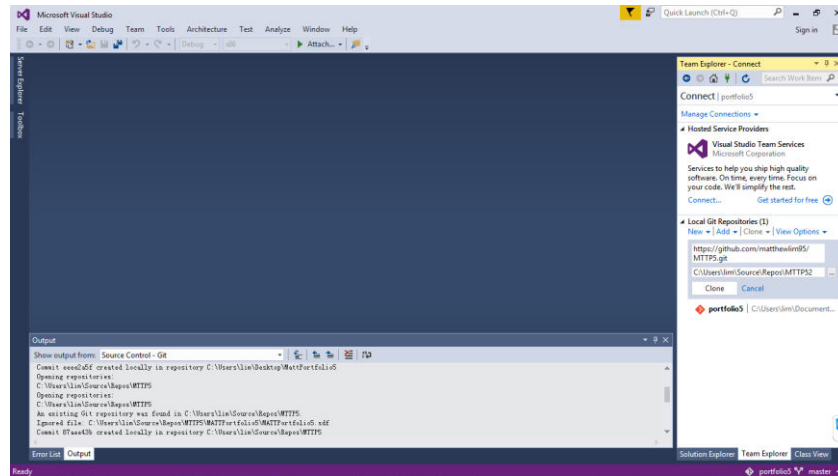
History – A list of changes made to a file since it was created.

Sync – Synchronize your files with the latest from the repository. This lets you grab the latest revisions of all files.

Revert – Delete your local changes and reload the latest version from the repository.

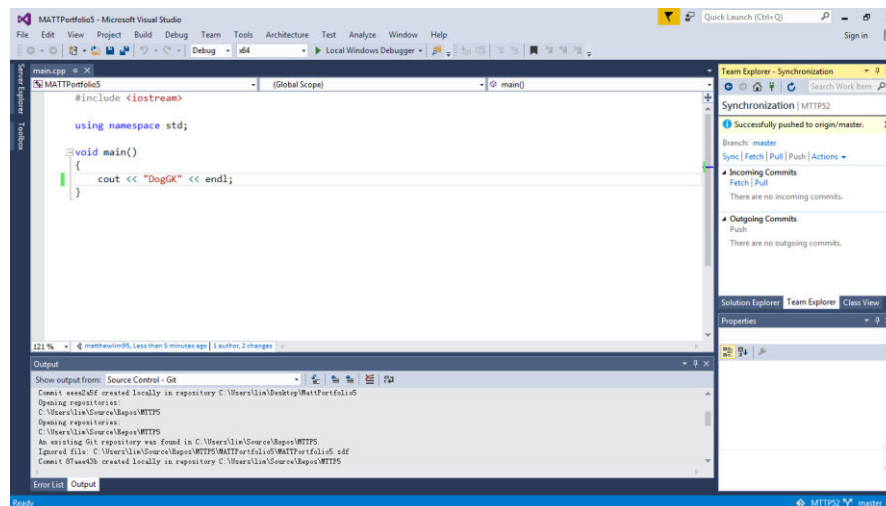
Merge – Apply the changes from one file to another, to bring it up-to-date.

Conflict – When pending changes to a file contradict each other. This usually happens when two or more contributors are changing a same file at the same time.

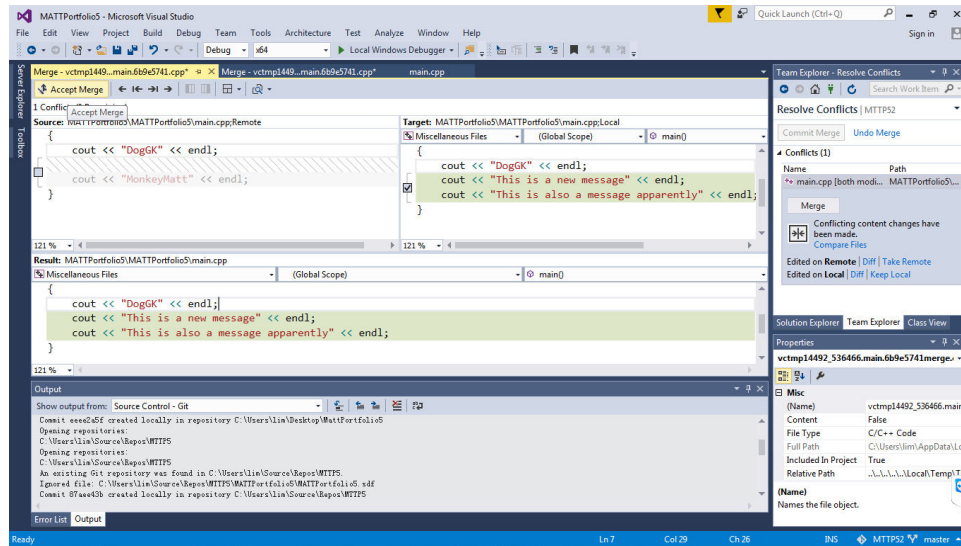


This evidence shows that a repository is being open in Microsoft Visual Studio. The pictured shown above describes the adding function where a repository is being added and check-in.

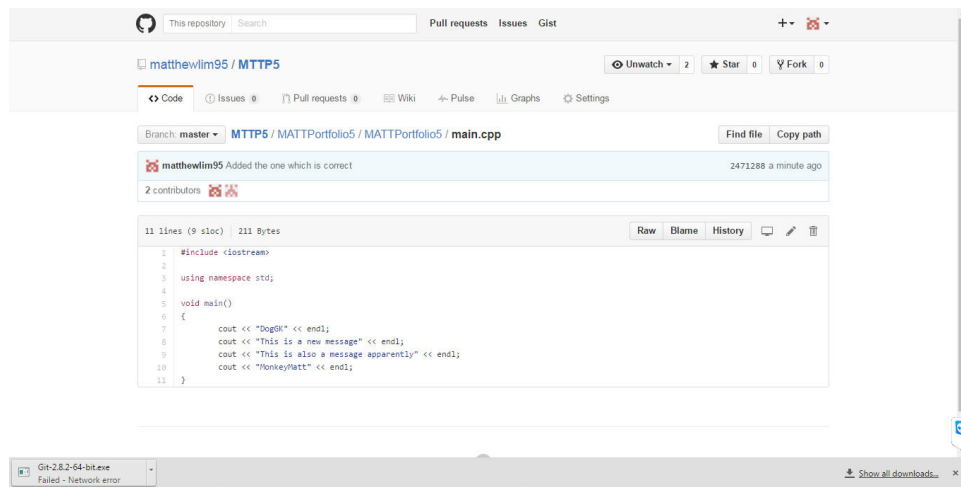
Many repositories can be worked on at the same time depending on how many collaborators are working on it. But the changes made by everyone that is working on the same file will be classified as a conflict.



The photo shown above records the activity when the master in GitHub pushes in a new file into the repository. Once the file is pushed into the repository, other collaborators are free to download the file and make changes to it.



The event that occurred in the evidence photo shown above is considered as a conflict, where two contributors are working on the same file. The master of the repository has to choose a more suitable and reliable changes made between the contributors. Once a conclusion has been made, the master will have to press the merge button to accept the selected changes.



This is the evidence of the sync function working on the repository created via GitHub. As the photo has shown, both updates from the previous merge have been included into the file. Whatever changes that has been made, once it is synced to the repository, all contributors will be able to download the file and continue on the project.