

Git Demystified

Wednesday, 14 October 2020 10:30 (2 hours)

Familiar with basic git commands, but not sure where to go next? Have you been copying mysterious git snippets from stackoverflow? This workshop is a deep dive into the surprisingly elegant underlying mechanisms that git uses to represent your code. We'll use this mental model to understand how git works and what it can do for us. The workshop only assumes basic knowledge of git, but some months of regular usage of basic commands (add/commit) will be an advantage.

Workshop Requirements:

The workshop assumes a basic familiarity with a basic git workflow for creating commits. You should be comfortable with using git add and git commit to create new commits.

In order to follow along with this workshop, you'll need an installation of a command line version of git which runs in a bash shell. Whilst the understanding you'll gain will be transferrable to your preferred git tools (e.g. GUI tools or editor integration tools), we recommend that you avoid the temptation to follow along with any tools other than the git command line in a bash shell.

For windows users, this means you'll need to install and run git bash. Instructions for doing so can be found in the Setup on Windows section below.

Setup on Windows:

- Download the Git for Windows installer.
- Run the installer and follow the steps below: Click on "Next" four times (two times if you've previously installed Git). You don't need to change anything in the Information, location, components, and start menu screens.
- *From the dropdown menu select "Use the nano editor by default" (NOTE: you may need to scroll up to find it) and click on "Next".*
 - Ensure that "Git from the command line and also from 3rd-party software" is selected and click on "Next". (If you don't do this Git Bash will not work properly, requiring you to remove the Git Bash installation, re-run the installer and to select the "Git from the command line and also from 3rd-party software" option.)
 - Ensure that "Use the native Windows Secure Channel library" is selected and click on "Next".
 - Ensure that "Checkout Windows-style, commit Unix-style line endings" is selected and click on "Next".
 - Ensure that "Use Windows' default console window" is selected and click on "Next".
 - Ensure that "Default (fast-forward or merge)" is selected and click "Next"
 - Ensure that "Enable Git Credential Manager" is selected and click on "Next".
 - Ensure that "Enable file system caching" is selected and click on "Next".
 - Click on "Install".
 - Click on "Finish" or "Next".
- If your "HOME" environment variable is not set (or you don't know what this is):
 - Open command prompt (Open Start Menu then type cmd and press Enter)
 - Type the following line into the command prompt window exactly as shown: `setx HOME "%USERPROFILE%"`
 - Press Enter, you should see SUCCESS: Specified value was saved.
 - Quit command prompt by typing exit then pressing Enter

This will provide you with both Git and Bash in the Git Bash program.

There is an excellent tutorial on setup for Windows here:

<https://youtu.be/339AEqk9c-8>

Setup on Mac OS

The default shell in some versions of macOS is Bash, and Bash is available in all versions, so no need to install anything. You access Bash from the Terminal (found in /Applications/Utilities). See the Git installation video tutorial for an example on how to open the Terminal. You may want to keep Terminal in your dock for this workshop.

To see if your default shell is Bash type 'echo \$SHELL' in Terminal and press the Return key. If the message printed does not end with '/bash' then your default is something else and you can run Bash by typing bash

If you want to change your default shell, see this [Apple Support article](https://support.apple.com/en-au/HT208050) and follow the instructions on "How to change your default shell".

For macOS, install Git for Mac by downloading and running the most recent "mavericks" installer from [this list](http://sourceforge.net/projects/git-osx-installer/files/). Because this installer is not signed by the developer, you may have to right click (control click) on the .pkg file, click Open, and click Open on the pop up window. After installing Git, there will not be anything in your /Applications folder, as Git is a command line program. For older versions of OS X (10.5-10.8) use the most recent available installer labelled "snow-leopard" [available here](http://sourceforge.net/projects/git-osx-installer/files/).

Setup on Linux

The default shell is usually Bash and there is usually no need to install anything.

To see if your default shell is Bash type `echo $SHELL` in a terminal and press the Enter key. If the message printed does not end with '/b

If Git is not already available on your machine you can try to install it via your distro's package manager. For Debian/Ubuntu run:

```
sudo apt-get install git
```

and for Fedora run:

```
sudo dnf install git.
```

Presenter: DAWSON, Mark

Session Classification: Parallel Session: Distributed programming