

## Source Control Method & Approach

**Summary** Source control is a central element in secure file storage, transfer, sharing and use. It is also really damn easy and quick when practiced right, cross platform too. The following doc illustrates this with the intent of promoting wide & uniform use.

**Author** Justin Reina

**Date** 7/30/17

### Platform Coverage

- Linux
- Windows
- Macintosh

### Software Selection

- terminal (Linux: tty / Win: Cygwin /
- git Mac: Terminal)
- gitk

### Repository Location

- Recommended & Jmr Use: 100% Local (.git/)
- Commonly Used: Remote Server (GitHub)

### Useful & Recommended

- \*Eclipse (Git & Team Viewer)
- \*Cygwin (Windows Bash Terminal)

Ask Justin for how to add to Right-Clicks

### Base Intent

- Track all revisions & changes
- Tag & track releases
- Tag & track development

### Example (Jmr, ASK Ref Project)

The example shown in Figure 1 illustrates:

- Tags(yellow) – tracking
  - e.g. 'r1' for rev 1, released to team
  - e.g. '6-28\_handoff' for last handoff
- Branches (green) – development
  - e.g. 'stat\_lib' for the statistics dev
- Form & Structure – type of commit
  - '(+)' – "Addition"
  - '(C)' – "Change"
  - '(B)' – "Bug"
  - '(M)' – "Misc."
  - '(U)' – "Update"
  - '(\*)' – "Unknown"

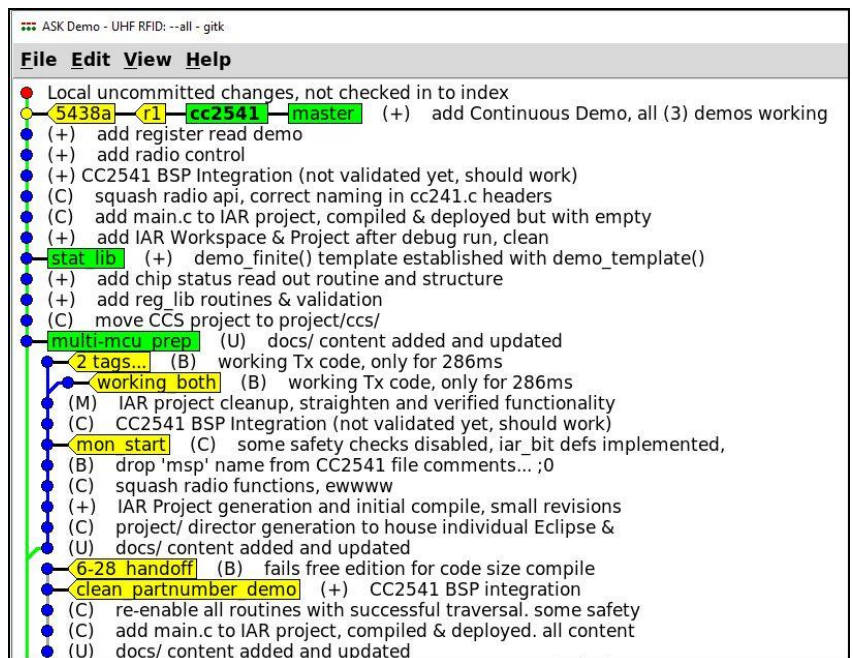


Figure 1: Example Repository

## Vocab

- Repo
- Commit
- Rebase
- Master
- Branch
- Tag
- HEAD

## Important Commands

Memorize these, to heart. This is 100% of what is needed to successfully implement & maintain source code control.

### Creation

- “git init”
- “git status”

### Generation

- “git add <file>”
- “git add \*”
- “git rm <file>”

### Commit

- “git commit”
- “git reset --hard” (reset to HEAD)
- “git reset --hard <commit-id>” (reset to a specific commit)

### Review & Correct

- “gitk --all &” (see Figure 1 for example. My primary repo viewer, quick & easy)
- “git commit --amend” (update an existing commit)
- “git rebase -i HEAD~1” (where ‘1’ is how far back you’d like to rebase)

## Useful Commands

These are used often, and they promote clean & organized repository development. Establish the habit early, and often!

### Tagging & Tracking

- “git tag *tag\_name*” (tag a commit with a tag, a name for later use & reference)
- “git checkout -b *branch\_name*” (checkout a new branch)

That’s it! More content will be provided here as needed or requested, please ask if interested (e.g. methodologies & procedure listings).