

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 7/9/17

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 X 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"

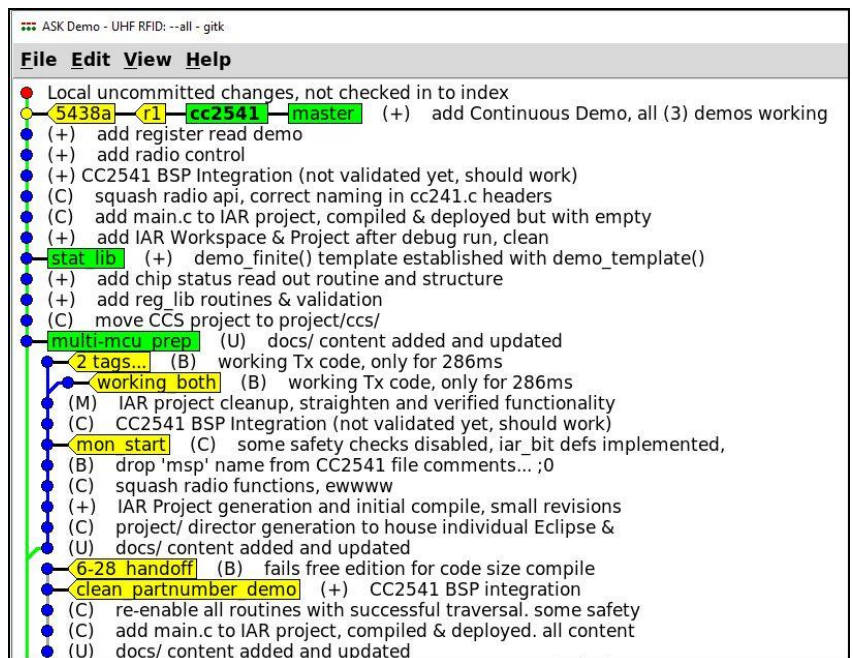


Figure 1: Example Repository

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 commit –amend”
- “git reset --hard”
- “git reset –hard <commit-specified>”

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