



Software Revision Control for MASS

Git Installation / Configuration / Use

Matthew Sell, CSSE Student
MASS Research Participant, February 2014



Overview

- Download / execute installer
- Initial configuration
- Clone repository

Which Git?

- Command-line (Windows, OS X, Linux)
 - <http://git-scm.com>
- GUI (Windows) : TortoiseSVN-like
 - <http://msysgit.github.io/>
 - <http://code.google.com/p/tortoisegit/>
- GUI (Windows, OS X)
 - <http://www.sourcetreeapp.com>
- Eclipse Egit plugin
 - “Team” menu adequate for: add/commit/push/pull
 - Use “History” view for complex operations

Installing Git “command-line”, Windows

- Download installer from: <http://git-scm.com>
- Run the installer, choose install folder
- Default components are acceptable
- Select “Run Git from the Windows Command Prompt”
- “Use OpenSSH”
- “Checkout Windows-Style, Commit Unix-Style”
- Reboot not required

Configuring Git “command-line”, Windows

- Open a command prompt (run “cmd”)
- `git config --global user.name “Your Full Name”`
- `git config --global user.email “your.email@address”`
- “Use Git Bash Only” option works fine
- “Use OpenSSH”
- “Checkout Windows-Style, Commit Unix-Style

Installing TortoiseGit, Windows

- Requires “command-line” Git – see slide #4
- Download installer from: <http://code.google.com/p/tortoisegit/>
- Run the installer
- Select “OpenSSH, Git default SSH Client”
- Default features are okay
- Reboot not required

Configuring TortoiseGit, Windows

- Make sure “command-line” Git was configured (slide #5)

Installing SourceTree, Windows

- Download installer from: <http://sourcetreeapp.com>
- Run the installer, choose destination folder
- When “Mercurial not found” dialog appears, select “I don’t want to...”
- Verify or enter your Full Name
- Verify or enter your Email address
- Enable “Allow SourceTree to modify your global ... config files”
- Enable “Configure automatic line ending handling by default”
- Agree to the license
- Select “Use PuTTY / Plink”
- No need to enter credentials for hosting services



Configuring SourceTree, Windows

- Already done – during installation

Installing Eclipse Plugin

- Click “Help”, “Eclipse Marketplace”
- In the “Find:” box enter “egit”
- Click “Install” in the “Egit – Git Team Provider” listing
- Click “Confirm” to confirm selected features – all three boxes checked
- This might be an update – click “Confirm” again if necessary
- Accept the license agreement, click “Finish”
- Click “Yes” when prompted to restart Eclipse

Using Git

- ✓ Clone remote repository to local
- ✓ Add new files
- ✓ Modify existing files
- ✓ Status / Diff
- ✓ Push / Pull / Conflicts
- ✓ Branching / Switching

The Repositories

- Practice repo
 - `ssh://dslab@hercules.uwb.edu:/~/Training/GitPractice.git`
- “Real” repos
 - `ssh://dslab@hercules.uwb.edu:/~/git/MASS_Java.git`
 - `ssh://dslab@hercules.uwb.edu:/~/git/MASS_CPP.git`
 - `ssh://dslab@hercules.uwb.edu:/~/git/MASS_CUDA.git`

Cloning (“checkout”)

- Git CL
 - Open command prompt in desired working directory
 - `git clone ssh://dslab@hercules.uwb.edu:/~/Training/GitPractice.git`
 - If that gives a “no address associated with name” error, remove the “:” after hostname
 - Enter DSLAB password
 - Will have a working directory (subdirectory) named “GitPractice”
- TortoiseGit
 - Open Windows explorer, right-click on desired working directory folder
 - Select “Git Clone”, enter URL (`ssh://dslab@hercules....`)
 - Click OK – note “Directory” matches “GitPractice”
 - Enter DSLAB password
 - Will have a working directory (subdirectory) named “GitPractice”
 - Click “Close”

Cloning (“checkout”)

- SourceTree
 - Click “Clone/New”
 - Enter “Source Path / URL” (ssh://dslab@hercules...)
 - Accept RSA key fingerprint (yes)
 - Click in “Destination Path” – SourceTree will attempt to verify connection
 - Accept server key, get some coffee...
 - Enter DSLAB password, can select “Remember password”, click “Login”
 - Should see “This is a Git repository”
 - Select “Destination Path” for working directory, leave other options default
 - Click “Clone” – repo will be cloned and will be listed in leftmost panel

Cloning (“checkout”)

- Eclipse Egit
 - Click “File”, “Import”, expand “Git” folder, click “Projects from Git”, click “Next”
 - Select “Clone URI”, click “Next”
 - Enter Host: hercules.uwb.edu
 - Enter Repository path: /~/Training/GitPractice.git
 - “Reselect” ssh
 - Enter User: dslab
 - Enter Password: (password) ... can select “Store in Secure Store” if you like
 - URI should have been constructed to match the URLs in previous slides
 - Click “Next”
 - May have to accept RSA key, .SSH directory creation dialogs
 - “master” branch should be selected, click “Next”
 - Select desired working directory, other options default, click “Next”
 - Select “Import as general project” (practice only!), click “Next”, click “Finish”

Quick review...

- Git tracks changes to your working copy
- New files must be “added” to the “index” / “staging area”
- “Push” commits to remote (“origin”)
- Each commit creates a “pointer” to a specific project state
- All repositories contain all project states (typically)
- Creating branches, moving between states is very quick
- Commit often. Branch often.
- Don’t commit breakage!
- VOE: Commit before pull.

Checking status

- CL Git
 - git status
 - Helpful hints provided for various activities
- TortoiseGit
 - Look at icons in Windows Explorer (may need to refresh)
- SourceTree
 - Look in “Working Copy Changes” panel
- Eclipse Egit
 - Look at icons next to filenames in Project Explorer

Adding Files to Index / Staging

- CL Git
 - `git add <filename>`
- TortoiseGit
 - Right-click filename, click “TortoiseGit”, then “Add...”
 - Click “OK” in confirmation dialog, Click “OK” again in second (!) dialog
- SourceTree
 - Highlight filename in “Working Copy Changes”, click the single ”up” arrow
- Eclipse Egit
 - Right-click filename, select “Team”, “Commit”
 - Enter commit message, select (checkmark) file in the “Files” panel, click “Commit”

Whoops! (Unstage!)

- CL Git
 - `git reset HEAD <filename>`
- TortoiseGit
 - Right-click filename, click “TortoiseGit”, then “Revert...”
 - Click “OK” in confirmation dialog, Click “OK” again in second (!) dialog
- SourceTree
 - Highlight filename in “Staged Changes”, click the single ”down” arrow in “Working Copy Changes”
- Eclipse Egit
 - Right-click filename, select “Team”, “Remove from Index”
 - Note: File still is shown as being in the index

Commit to Index (Local)

- CL Git
 - git commit
 - Add commit message (starts Vi editor)
- TortoiseGit
 - Right-click in directory, click “TortoiseGit”, then “Git Commit ->”
 - Add commit message, make sure correct files check marked in bottom list
 - Click “OK”, then “Close”
- SourceTree
 - Click “Commit” button in top toolbar, enter commit message, click “Commit”
- Eclipse Egit
 - Right-click filename, select “Team”, “Commit”
 - Enter commit message, select (checkmark) file in the “Files” panel, click “Commit”

Whoops! (Revert!)

- CL Git
 - Git revert HEAD
 - Enter revert commit message
- TortoiseGit
 - Right-click directory, click “TortoiseGit”, then “Show Log”
 - Last commit highlighted in upper list (graph), “Revert Change By This Commit”
 - Click “OK” in both confirmation dialogs
- SourceTree
 - Double-click prior commit listing in revision history/graph, select “Clean (discard all changes)” in “Confirm change working copy”, Click “OK”
- Eclipse Egit
 - Right-click project, select “Team”, “Show in History”
 - Highlighted line is latest commit
 - Right-click highlighted line, click “Revert Commit”

Ignoring files

- CL Git
 - Add filename to “.gitignore” file
- TortoiseGit
 - Right-click directory, click “TortoiseGit”, then “Add To Ignore List”
 - Click on the complete filename, or a pattern
 - Click “OK” in confirmation dialog
- SourceTree
 - Highlight file in “Working Copy Changes”
 - Right-click, then click “Ignore...”
 - Choose desired pattern, then click “OK”
- Eclipse Egit
 - Right-click project, select “Team”, “Ignore”

Push To Remote (Pull first!)

- CL Git
 - git pull (may have to provide password)
 - git push origin master (“master” is the branch)
- TortoiseGit
 - Right-click directory, click “TortoiseGit”, then “Pull”
 - Make sure branch selections appropriate, click “OK” (password), then “Close”
 - Right-click directory, click “TortoiseGit”, then “Push”
 - Make sure branch selections appropriate, click “OK” (password), then “Close”
- SourceTree
 - Click “Pull” in toolbar, verify branch names, click “OK”
 - Click “Push” in toolbar, verify branch names, click “OK”
- Eclipse Egit
 - Right-click project, select “Team”, “Pull”, then “OK”
 - Right-click project, select “Team”, “Push to Upstream”, then “OK”

Quick review...

- Commit after a group of logical operations
- Must commit after delete, move, rename, etc.
- “Ignore” IDE configuration files, build output directories, compiled class files, generated artifacts, etc.
- Partition project responsibilities to prevent merge collisions – multiple people shouldn’t be working on the same files at the same time!
- Don’t forget to push to Remote!

Questions ???

Credits:

- Cherie Wasous: Slide Templates
- Jeff Meyer, Fluke Corporation: Git books and practical use
- Curt Mills, Fluke Corporation: Practical Git use
- Atlassian: SourceTree software, Git presentations, cheat sheet
- Roger Dudler: Simple Git examples, cheat sheet
- Scott Chacon, Apress: Pro Git book