# SIAM: Getting Started with Git

based on http://git-scm.com/book and slides by Bart Trojanowski

Andrew Reisner and Nathan Bowman

March 11, 2014



#### Table of Contents

- 1 Overview
- 2 Components
- 3 Operations
  - Creating/Updating
  - Getting Information
  - Branching/Remotes
- 4 Sharing
  - Distributed Workflows
  - Git on the Web





#### Overview



#### Git is a

- Free and Open Source
- Distributed
- Version Control System.







### Version Control System

#### Preserve a clear, timely record of software evolution

- Record changes to files
- History can be recalled/inspected

#### Implications:

- Rollback changes
- Know what collaborators are working on
- Investigate changes when bugs emerge
- Find how and where a particular bug was fixed





### Components



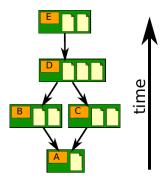
# VCS Components (Working Tree)

- Single checkout of one version of the project
- Directories
- Files



# VCS Components (Repository)

- Files
- Commits
- Ancestry



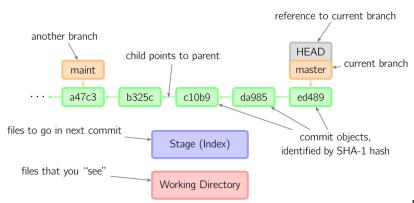




# VCS Components (References)

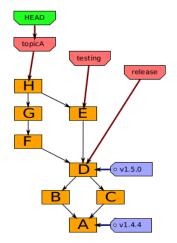
- Tags
- Branches

- HEAD
- Index (Staging area)





## VCS Components (Example Tree)







# Operations





### VCS Operations

#### Bootstrap

- init
- clone
- checkout

#### Modify

- add, delete (rm)
- rename (mv)
- commit

#### Information

- status
- diff
- log

#### Reference

- tag
- branch

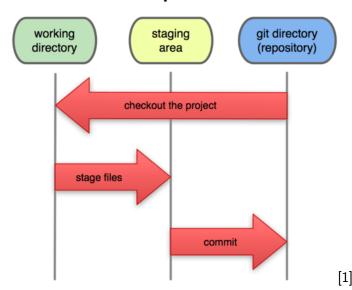
Sharing work, backing it up

- pull, fetch
- push





#### **Local Operations**





### **Bootstrapping**

- \$ git init
  - creates .git directory and initializes the repository
- \$ git clone <URL>
  - replicates a remote repository
  - checks out new working tree
  - Git URLs
    - /home/user/my-project.git
    - http://github.com/user/my-project.git
    - git://remote.server/my-project.git
    - user@remote.server:my-project.git
    - ssh://user@remote.server/ user/my-project.git





### Staging

- \$ git add <path>
  - Adds contents of <path> to index
  - \$ git add .
- \$ git rm <file>
  - Removes files from working tree and index
- \$ git mv <source> <destination>
  - Moves or renames a file or directory
- .gitignore
  - Text file that specifies files to ignore





# Example .gitignore file

- \*.aux
- \*.fdb\_latexmk
- \*.fls
- \*.nav
- \*.out
- \*.snm
- \*.toc
- \*.vrb
- **↓** ~





### Changing Settings

- \$ git config --list
  - Lists the current configuration settings
- \$ git config <key>
  - Gets the current value of key
- \$ git config [level] <key> <value>
  - Changes setting key to value
  - Optional level determines scope of setting
    - Omitting level: repository
    - --global: user
    - --system: system





### Common Configuration Settings

A few settings you will want to update when first using Git:

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
$ git config --global core.editor emacs
$ git config --global core.excludesfile ~/.gitignore
$ git config --global merge.tool meld
```





```
$ git commit -m <msg>
```

- Creates a commit of staged items
- \$ git commit -m "fixes issue #108"





- \$ git status
  - Displays the working tree status
  - staged, unstaged, untracked
- \$ git diff
  - Displays changes between index and working tree
- \$ git diff --staged
  - Displays changes between HEAD and index
- \$ git diff HEAD
  - Displays changes between HEAD and working tree
- \$ git diff <commit> <commit>
  - Displays changes between two commits



### Demonstration of Staging

```
$ echo "foo" >> myfile
$ git diff myfile
diff --git a/myfile b/myfile
index e69de29..257cc56 100644
--- a/myfile
+++ b/myfile
@@ -0,0 +1 @@
+foo
```



## Referencing Objects

- a88dbbe57b9e9fc01f701c45c405647c588e6a6a
- a88d
- v1.0.3
- master
- origin/master
- HEAD
- HEAD<sup>^</sup> == HEAD<sup>^</sup>1
- feature\_brach@{May.30}





### Show and Log

- \$ git show <object>
  - Show various types of objects
  - \$ git show HEAD@{yesterday}
  - \$ git show HEAD:file
- \$ git log [<since>...<until>] [-- <path>]
  - Show commit logs
  - \$ git log HEAD~3..HEAD^
  - \$ git log -- file-with-bug.c





### Log Formatting

```
$ git log --pretty=<format>
```

- oneline
- full
- format: "hash: %h author: %an date: %ad"
- see git-log(1) for more options
- \$ git log --graph --pretty=oneline





### Log Formatting - Tree Example

```
edf4f64dl1875eaf Git: adds bibliography
    b213ffe8782b7fc0a3ffad57aad7ba024f2bd7b0 Merge remote-tracking branch 'origin/master'
       bb93fcbc30b375aa33c7db873dae873e15ac34d6 Merge branch 'master' of https://github.com/SIAM-at-Illin
        dd4540f084944dce055e6b16e8301502a552cae3 minor additions to latex slides
458cla3lcdacld1027a0d74291f4beb7e279391b adds Git naming clarification slide, improves Git compone
        a4edal37cce3eleb22093cb25d7fd80ff97llf06 Merge remote-tracking branch 'origin/master'
    720522957b8459004ea8e81ff049631761092c62 minor additions to latex slides
        2a8a9668d65149705943a3a703ba77a6c7c88a69 Merge remote-tracking branch 'origin/master'
//23:field=07bblcs08bcd3d1440b08bc08680c0 R: fixing a couple of examples
Basinbod no body of the couple of examples
Basinbod no body of the couple of examples
Basinbod no body of the couple of the couple of examples
Basinbod no body of the couple of the 
9a3663dc923c16a189fbd185c577d28a7d83e7fc adds more command slides and diagrams for conceptual slides
   2fb57fc073bc8545ade56b98f6c96lee22l15dal adding to latex slides
1670c.cde:0317.22013c.6d65eb03a0124d865a1 adds Git command overview slides

BB60bb2570s05 fol170163c4505654751B80a2 adds forgatten GitHub log

261352d97a013d212e093d459b6a47dc408c272dds working on lates slides

164852501947222586e68252887375c097fa055 Merge remote tracking branch 'origin/master'

- M8032501947222586e68252887375c097fa055 Merge remote tracking branch 'origin/master'
* 51a95fab5c8d5ee16a5c4863a31e144ada61294b adding to latex slides
       89498612021e784e23a376812719abdd69a0dc41 Merge branch 'master' of https://github.com/SIAM-at-Illin
        6af880d0d4d91710235bd62b343655lea4de34c1 updated latex presentation 250ae77ffbc6f393b3fc665c073c48683ae05548 adds content for Github slides, fixes spelling errors
        e2e9788cf2c975f48e7e87098fa314b2lf9f2435 Merge remote-tracking branch 'origin/master'
        d80d8d0ff6af996c59de82ca69ddlbddfa8fa6b9 R: plotting exercise
         98cadfb5554e1d529816f8fb2lc57f382c9c161a R: r-bloggers resource
42903f8c1068cc2955ccfc4c007dc6ff564ef478 R: web and editors resources
        91a58e24b1000bd280a9df121f218f34ea2a2293 R: recommended books
```





#### Branching

- \$ git branch -1
  - List local branches
- \$ git branch <branchname>
  - Create new branch on HEAD
- \$ git branch <branchname> <start-commit>
  - Create new branch on specified commit
- \$ git checkout <branch>
  - Checkout branch by name
- \$ git checkout -b <branchname> [<start-commit>]
  - Create and switch to a new branch





#### \$ git merge <branch>

- Incorporates changes from the specified branch into the current branch.
- Conflicts may result
- Any conflicts must be resolved before merge is completed

```
var = 3
<<<<<< HEAD
x = 0.5 * var
======
x = 1/2. * var
>>>>>> origin/master
```

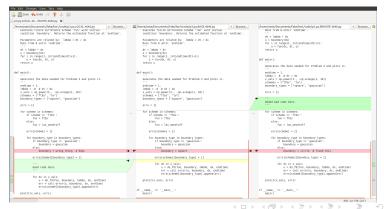




Overview Components Operations Sharing Creating/Updating Getting Information Branching/Remotes

### Mergetool

- \$ git merge <branch>
  - Presents a visual interface to merging
  - Example:
  - \$ git mergetool --tool=meld







#### Remotes

- \$ git remote add <name> <url>
  - Adds a remote named <name> for the repository at <url>
- \$ git fetch <remote>
  - Fetches updates from specified remote
  - \$ git fetch --all
- \$ git branch -r
  - List remote branches
  - Use \$ git merge to merge these branches
- \$ git pull [<remote>] [<branch>]
  - Short for a fetch followed by a merge



# Git Naming-Disambiguation

Git creates branches automatically in certain cases.

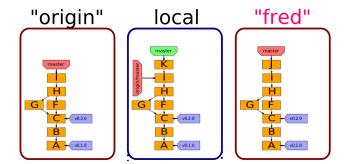
- HEAD: special reference that identifies the current branch
- master: Default branch created when a repository is first initialized
- origin: default name chosen for a remote when cloned
- <remote\_name>/<branch\_name>
  - origin/master
  - upstream/fix-issue-105





### Remotes Example

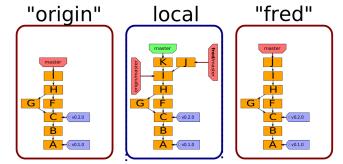
"fred" cannot push to "origin"





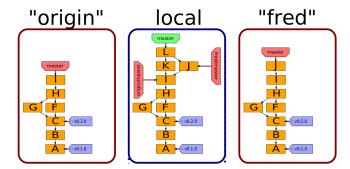
# Remotes Example (continued)

Fetch from "fred"



# Remotes Example (continued)

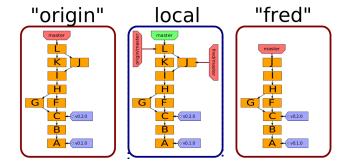
#### Merge in the changes





Remotes Example (continued)

Push changes to "origin"





#### Challenge Problem

Shape module at https://github.com/gswg/example.git

- Clone repository
- Locate and fix bug
- Push fix
  - you may need to fetch and merge with origin/master
  - username: gswg
  - password: siam2014





# Sharing





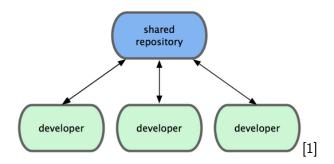
#### Distributed

- No central location that keeps track of your data (no single place is more important than another)
- Encourages small commits and frequent merging
- Branches don't affect the main repository and can commit changes without disturbing others
- Work offline
- Rely on a network of trust





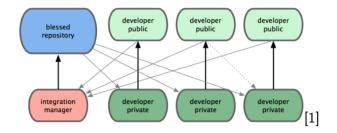
#### Centralized Workflow







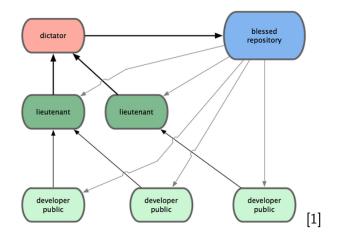
### Integration-Manager Workflow







#### Dictator and Lieutenants Workflow







### Free and Open Source

- Downloads at http://git-scm.com
- Libgit2: free and open source library for writing custom Git applications









#### GitHub

- Powerful web interface for publishing Git repositories
- Simple to view changes and track progress on repositories
- Wiki and bug tracking built into each repository







- Similar to GitHub
- Allows private repositories for students







#### References

- Git Book. URL http://git-scm.com/book.
- [2] Git From the Bottom Up. URL http://ftp.newartisans.com/pub/git.from.bottom.up.pdf.
- [3] Git Magic. URL http://www-cs-students.stanford.edu/~blynn/gitmagic/.
- [4] User Manual. URL http://git-scm.com/docs/user-manual.html.
- [5] Code School Try Git. URL http://try.github.io.
- [6] Tech Talk: Linus Torvalds on Git. URL http://youtu.be/4XpnKHJAok8.
- [7] Mark Lodato. A Visual Git Reference. URL marklodato.github.io/visual-git-guide/.
- [8] Bart Trojanowski. Bart's Blog-Intro to Git. URL www.junkie.net/~bart/blog.



