

USING GIT ON OS-X

Munich Cocoaheads 2009-11-12

©2009 Stephen Riehm

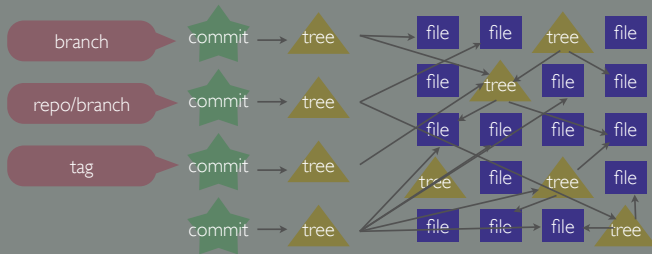
WHAT IS GIT?

project_dir/

.git/

INSIDE GIT

.git/



alice = ssh://alice.com:/git/repo.git

bob = git://bobco.com:/repo.git

BLOBS

569061c1725109d441fe846300fd4e57063cc6d6b

blob	size
<pre>// Blah.m // This class does @implementation Bla @synthesize a;</pre>	

TREES

56906c1...

tree		size	
rw-rw-rw-	blob	5b1d3...	ReadMe
rwxtwxrwxtwx	tree	03e78...	Classes
rwxtwxrwxtwx	tree	cdc8b...	Resources
rw-rw-rw-	blob	cba0a...	Info.plist
rwxtwxrwxtwx	blob	911e7...	distrib.pl

COMMITTS

56906c1...

commit	size
tree	c4ec5...
parent	a149e...
author	Scott
committer	Scott
date	2009-11-10 21:15
commit message	

TAGS

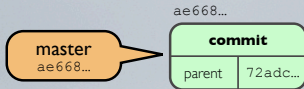
56906c1...

tag	size
object	ae668...
type	commit
tagger	Scott
tag message	

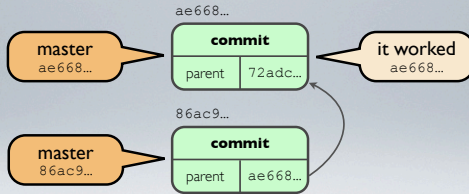
LIGHT-WEIGHT TAGS



BRANCHES



BRANCHES 'V' TAGS



THE WHOLE LOT

98ca9...

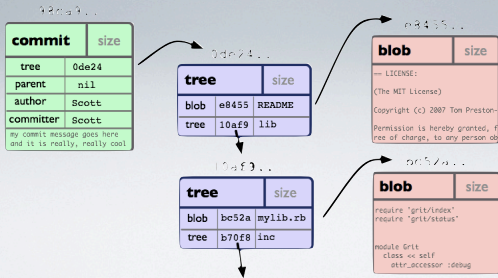
0de24...

10af9...

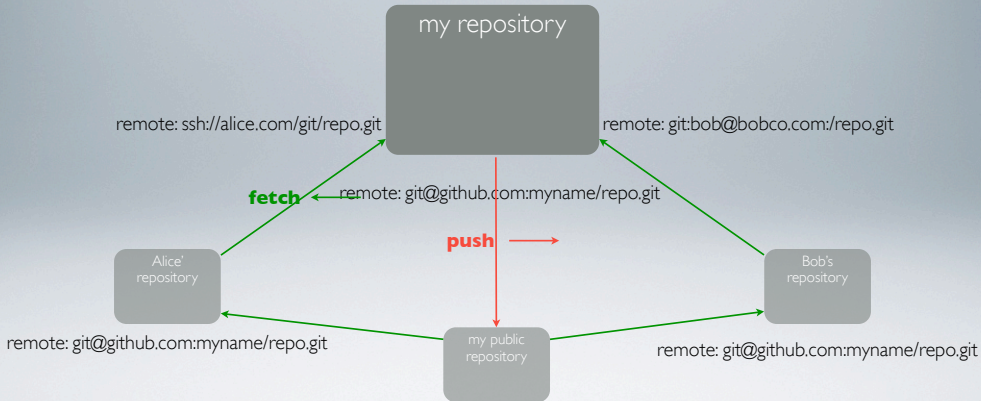
a7048...

bc52a...

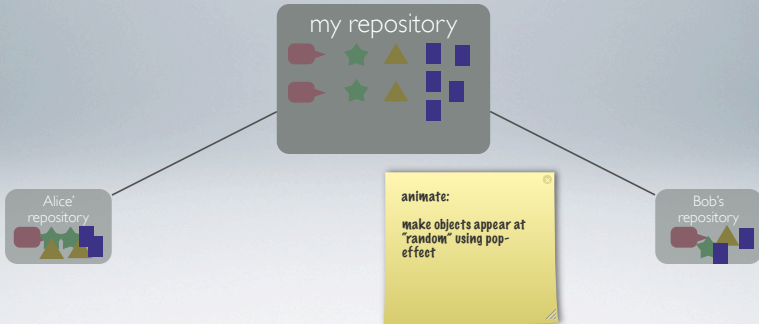
b70f8...



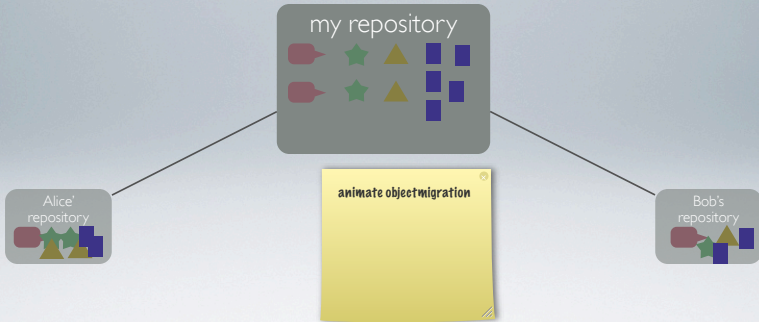
SHARING PROJECTS



NO INTERFERENCE



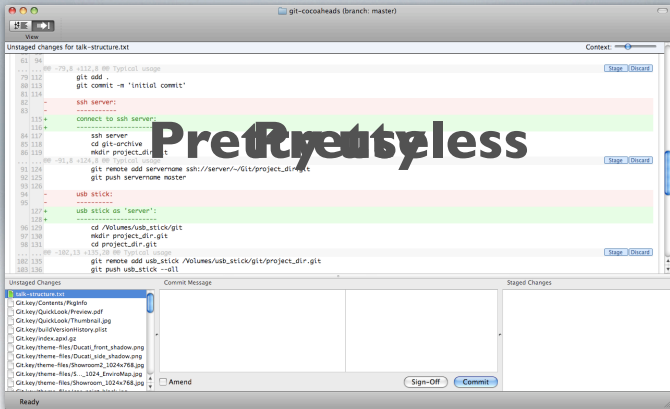
GOOD INTERACTION



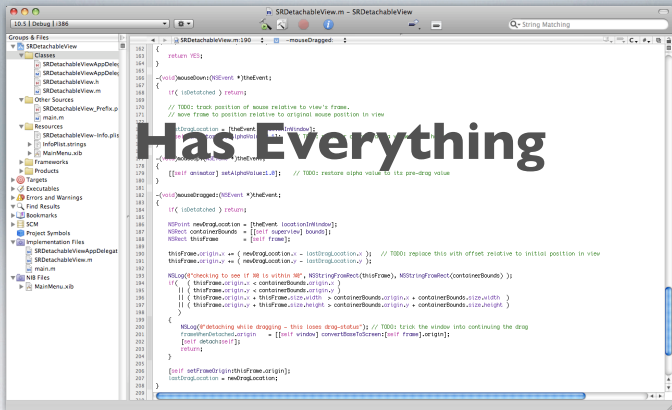
GIT YOUR HANDS DIRTY

```
~/ > echo "you're going to need the command line :-)"
```

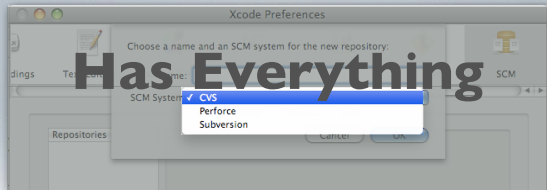
GITX



XCODE



XCODE



GLOBAL CONFIGURATION

```
02024]~/ > $EDITOR ~/.gitconfig
    pager = more
    excludesfile = /Users/me/.gitignore
[user]
    name = My Full Public Identity
    email = h4x0r@example.com
[format]
    pretty = format:%h %ci [%aN] %s
```

DAILY GIT

TYPICAL USAGE

WHENEVER I GET LOST

```
~/ > git help <cmd>
```

```
~/ > git help add
```

- <http://git-scm.com/documentation>
- <http://google.com> ;-)

TYPICAL USAGE

- Create a new XCode project
- Open Terminal

```
02:26 ~/ > cd <MyProject>
```

- Create an empty git repository

```
02:27 ~/ > git init
```

- Add your project files to the git repository

```
02:28 ~/ > git add .
```

- Set an initial label

```
02:29 ~/ > git commit -m 'initial commit'
```

TYPICAL USAGE

SETUP REMOTE REPOSITORY

- Connect to ssh server

```
02:30 ~/ > ssh me@myserver
```

- Go to your git projects folder

```
02:31 ~/ > cd git-archive
```

- Create a new folder for your project

```
02:32 ~/ > mkdir project_dir.git
```

```
02:33 ~/ > cd project_dir.git
```

- Create a bare repository

```
02:34 ~/ > git init --bare
```

- Switch back to your local host and add the remote location

```
02:35 ~/ > git remote add servername  
ssh://server/~/.Git/project_dir.git
```

- Push your project to the remote server

```
02:36 ~/ > git push servername master
```

TYPICAL USAGE

SETUP USB-STICK REPOSITORY

- Go to your git projects folder on your usb-stick

```
02:41 ~/ > cd /Volumes/usb_stick/git
```

- Create a new folder for your project

```
02:42 ~/ > mkdir project_dir.git
```

```
02:43 ~/ > cd project_dir.git
```

- Create a repository

```
02:44 ~/ > git init --bare
```

- Switch back to your local project directory

```
02:45 ~/ > cd ~/me/<local_project_dir>
```

```
02:46 ~/ > git remote add usb_stick  
/Volumes/usb_stick/git/project_dir.git
```

- Push your project to the usb-stick

```
02:47 ~/ > git push usb_stick master
```


TYPICAL USAGE

WHERE I AM CONNECTED TO?

- Find remote repositories for your current project

```
02:50 ~/ > cd ~/me/<local_project_dir>
```

```
02:51 ~/ > git remote -v
```

```
server_alias ssh://remote_server/remote_project_dir/project.git
```

```
usb_alias    /Volumes/usb_stick/project_dir/project.git
```

TYPICAL USAGE

SOURCING FROM MULTIPLE REPOSITORIES

- Go to your local project

```
02:60 ~/ > cd ~/me/<local_project_dir>
```

- Open git project config

```
02:61 ~/ > cd .git
```

```
02:62 ~/ > $EDITOR config
```

- Set up a repo-group

```
...
```

```
[remotes]
```

```
    group_name = servername1 servername2
```

```
...
```

TYPICAL USAGE

CLONE AN EXISTING PROJECT

- Go to your local working directory

```
03:00 ~/ > cd ~/development
```

- Get the project from remote server

```
03:01 ~/ > git clone -o servername URL
```

- Go to your project directory

```
03:02 ~/ > cd project_dir
```

- **Todo:** Describe this step

```
03:03 ~/ > git checkout -b my_stuff
```

- Or just "git checkout tag" if I'm building git, perl etc

TYPICAL USAGE

CLONE A LOCAL REPOSITORY

- Go to your local working directory

```
03:00 ~/ > cd ~/development
```

- If I want to see what git does

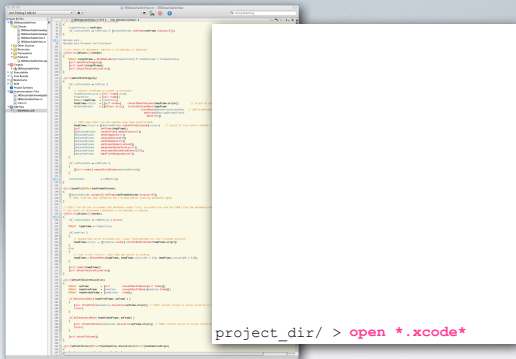
```
03:01 ~/ > git clone other_git_dir
```

TYPICAL USAGE

WORKING WITH OTHERS

- Each has their own published repositories
- NEVER push to anyone else's repository
- use "git remote update" to scan all friends' repos

XCODE & TERMINAL



DON'T PANIC



Alert

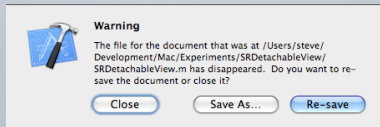
The project "SRDetachableView.xcodeproj" was changed externally. Do you want to keep using this version, or the one on the disk?

Keep this Version

Read from Disk

just do it...

DON'T PANIC



Maybe you should...

close XCode

checkout the version you want

open XCode again