Git 'R Done
An intro to Git Source Control Management

Based on slides by Ryan Felton of SD Ruby

Buzz

- Rails Core has moved to GitHub and Lighthouse

- Rubyforge supports it

- Piston supports it

What makes Git special?

Git was designed from the ground up as a distributed version control system. Being a distributed version control system means that multiple redundant repositories and branching are first class concepts of the tool.

Installation

- Mac OS X

  - git-osx-installer

- Ubuntu Linux

  - sudo apt-get install git git-core git-svn

- Windows

  - Cygwin

- Else

  - Install from source

The Basics

GIT VS SUBVERSION

SVN Benefits

- Single Repository

- Access Control

- Shorter Revision Numbers

- We know how to use it

Git Benefits

* Distributed Nature

* Branch Handling

* Performance (Speed)

* Repository Size

* Powerful (at the cost of simplicity)

SVN Branches

- svn copy http://www.rubyonrails.org/svn/trunk http://www.rubyonrails.org/svn/branches/myfix

- svn checkout http://www.rubyonrails.org/svn/branches/myfix

- (CHANGES)

- svn merge http://www.rubyonrails.org/svn/branches/myfix

- svn checkout http://www.rubyonrails.org/svn/trunk

- svn commit -m "merged myfix branch to trunk"

Git Branches (Local)

- git branch myfix

- git co myfix

- git co master

- git merge myfix

- git push

What do you do when...?

1) You are half way done with a new feature and an urgent bug comes up. You need to fix the bug and commit it immediately.

2) You just did submitted a commit and realize that you forgot to...

GO!

1) You are half way done with a new feature and an urgent bug comes up. You need to fix the bug and commit it immediately.

a) If the bug is in a file unaffected by your changes...

‣ Write the fix and commit just that file

b) If the bug is in a file you are currently working on (small)...

‣ Stash your changes, make the change, commit, unstash

c) If the bug is large and will take some time to fix...

‣ Create a new branch for your feature, leave your sweet new code in the branch, return to master branch and fix bug.

2) You just did submitted a commit and realize that you forgot to…

a) You could hit undo and redo the commit

   1. git reset --soft HEAD^

   2. edit

   3. git commit -a -c ORIG_HEAD

b) Or you could just add onto the last commit

   1. git commit --amend

git revert != svn revert

SVN Revert

- Destroys uncommited local changes

- Restores to the latest commit

Git Revert

- Local changes will still be present

- Reverts the last COMMIT (destructive)

Solution

git reset --hard = svn revert .

Migrating from SVN?

git-svn

- git-svn allows you to use git on your local machine (even offline) and push your commits to your subversion repository

- Preform normal git commits and branches

- git svn rebase to "pull" latest updates

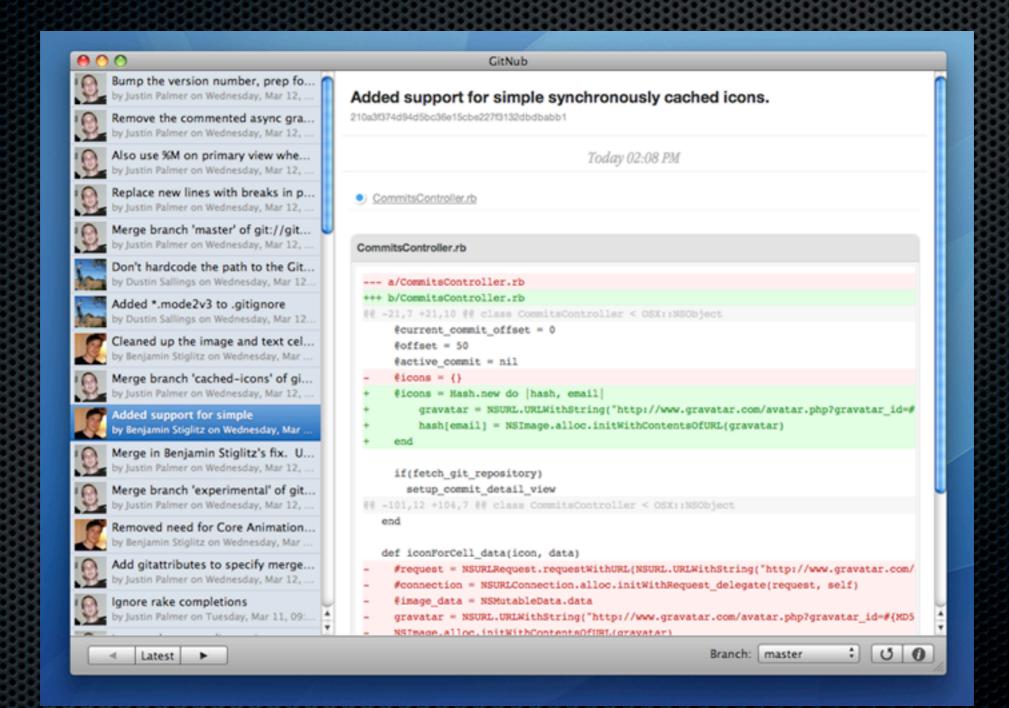- git svn dcommit to "push" your local commits

gitify

```ruby
1   #!/usr/bin/env ruby -wKU
2   # Based on http://drnicwilliams.com/2007/11/22/going-offline-without-your-favourite-subversion-repository/
3
4   # get svn info location
5   if ARGV.empty?
6     svnurl = `svn info | grep "^URL:"`.gsub('URL: ','').chomp
7
8     # project = basename
9     project = "../#{File.basename(Dir.pwd)}.git"
10  elsif ARGV[0] =~ %r{^svn(\+ssh)?://} || ARGV[0] =~ %r{^http://}
11    svnurl = ARGV[0]
12
13    if ARGV[1].nil?
14      project = svnurl.split("/").last
15
16      project = svnurl.split("/")[-2] if project == "trunk"
17    else
18      project = ARGV[1]
19    end
20  else
21    puts "#{ARGV[0]} does not appear to point to a SVN Repo"
22    exit
23  end
24
25  puts cmd = "git svn clone #{svnurl} #{project}"
26
27  `#{cmd}`
```
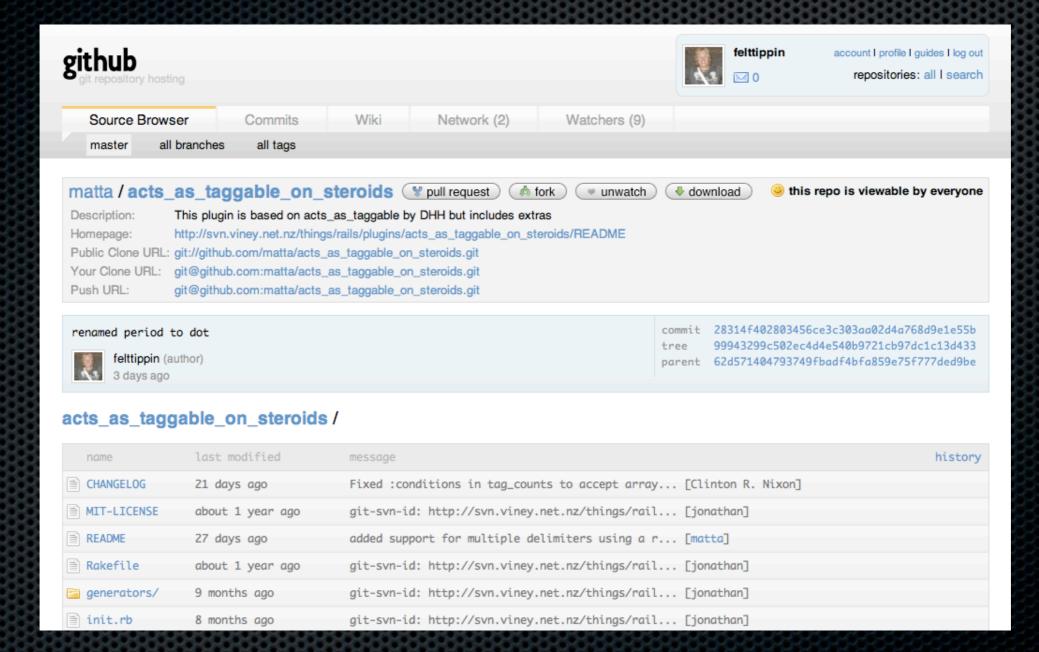
http://pastie.caboo.se/148312

Resources

GUI: GitNub

Hosting: GitHub

Distributed Revision Control Systems

- Git

- Mercurial

- Bazaar

Questions?

Rob Olson

   College Student

   Jobless Ruby Programmer

rko618@gmail.com

http://thinkingdigitally.com