

An Introduction to Git

Simple Distributed Revision Control

Austin Godber

godber@uberhip.com

<http://uberhip.com/projects/presentations.html>



May 28, 2008

Introduction

Git Introduction

- Simple Stand Alone Revision Control
- Powerful Distributed Revision Control
- Easy Public Repository Sharing
- Cross Platform Availability

Git Installation

- **Windows** - msysGit
<http://code.google.com/p/msysgit/>
- **OS X Leopard** - Git OS X Installer
<http://code.google.com/p/git-osx-installer/>
- **Linux** - apt-get/yum install git-core

Git Extras

- Hosted Git Services
- Personal Git Server, ssh or git-server
- Web Based Repository Browsing
- Git UI Tools
- Git based backup

Basic Usage

Setup and Initialization

Start off making a few setting a few global variables:

```
git config --global user.name 'Austin Godber'  
git config --global user.email godber@uberhip.com  
git config --global color.status auto  
git config --global color.branch auto
```

You can create a git repository in the current project directory by:

```
git init  
git add .  
git commit -m "initial commit of project"
```

Basic Workflow

Make some changes, review what you have done, stage (add) those changes, commit:

```
# make some changes to README.txt
git status
git diff
git add README.txt
git commit -m "updated the description"
```

Make some BAD changes, revert to the last state:

```
# make some bad changes to config/hoe.rb
git status
git reset --hard HEAD
# NOT -> git revert HEAD
```

Ignoring and Digging for info

There are two ways to ignore files with git:

- **Project Global** - edit the `.git/info/exclude` file
- **Per Directory** - put it in a `.gitignore` file

Sift through commit messages with **git log**:

```
git log
git log --since="2008-05-20"
git log --since="1 week ago" --until="2 days ago"
git log path/to/file
```

Simple Branching and Merging

checkout, branch, pull and merge

```
# write a Hello World program tmp/hw.rb
git add tmp/hw.rb
git commit -m "hello commit"
git checkout -b refactor
# refactor out our puts
git commit -a -m "refactored puts"
git branch                # lists branches
git checkout master
git pull . refactor      # fast forward merge
```

Use **git stash** to hold onto changes. Conflicts are marked with “<”. Resolve them then **git add** the files.

Gitk

The screenshot shows the Gitk application window titled "gitk: git". The window has a menu bar with "File", "Edit", and "View".

Commit History:

- **master** full outline Austin Godber <godber@uberhip.com> 2008-05-24 10:47:38
- presentation outlined and introduction started Austin Godber <godber@uberhip.com> 2008-05-22 08:14:04
- initial stub out of Git presentation Austin Godber <godber@uberhip.com> 2008-05-20 23:32:33

SHA1 ID: ef5430f080e61214fa8f08ef381a9a941e66e6dc

Highlight: Commits touching paths: OR in view None OR None

Diff View:

Author: Austin Godber <godber@uberhip.com> 2008-05-22 08:14:0
 Committer: Austin Godber <godber@uberhip.com> 2008-05-22 08:1
 Parent: cfa25f9bd2579340edfac7592305d47d8d1c0df (initial stub
 Child: 5538738d9677a9cf7522a62622817b4e4f845ca7 (full outline
 Branch: **master**
 Follows:
 Precedes:

presentation outlined and introduction started

git-Intro.tex

```

index 9531a4d..1ea88a 100644
@@ -66,14 +66,32 @@
 \section{Introduction}

 \frame{\frametitle{Git Introduction}
-
+ \begin{itemize}
+ \item Simple Stand Alone Revision Control
+ \item Powerful Distributed Revision Control
+ \item Easy Public Repository Sharing
+ \item Cross Platform Availability
+ \end{itemize}
  }
  
```

Comments: Git-Intro.tex

Sharing Your Repository

Git Server - You already have it!

Git comes with a simple web based repo browser.

```
git instaweb --httpd=webrick  
git instaweb --stop
```

You can edit `.git/description` and check out `.git/gitweb` for customization.

Simple Read Only Sharing over HTTP

Make the WWW directory and initialize a git repository there:

```
mkdir -p ~/Sites/git/gitintro.git
cd ~/Sites/git/gitintro.git
git --bare init
```

Configure your repo to push there, in your repo dir:

```
git remote add public ~/Sites/git/gitintro.git
git push public master
```

Make one last tweak in the WWW dir:

```
cd ~/Sites/git/gitintro.git
git --bare update-server-info
chmod a+x hooks/post-update
# now you can:
# git clone http://localhost/~godber/git/gitintro.git/
```

Additional Git Resources

Git Repository for this presentation:

```
git clone http://uberhip.com/git/gitintro.git
```

Austin's Git Related Links:

```
http://del.icio.us/godber/git
```

Thank You

Austin Godber

godber@uberhip.com

<http://uberhip.com/projects/presentations.html>

JumpBox 

Colophon

Presentation produced using the Beamer class in \LaTeX using the Kile and Vim editors on Ubuntu 7.10.