Git going with DVCS

60 minutes of source code control magic
Hello my name is Matthew McCullough
http://delicious.com/matthew.mccullough/git
git
“cool kids” version control system
Open Source
“Git

-noun

British Slang. an unpleasant or contemptible person”

-Oxford English Dictionary
“I’m an egotistical bastard, and I name all my projects after myself. First Linux, now git.”

-Linus Torvalds
Coming from Subversion?
Coming from Subversion?
✓ Coming from Subversion?

? Tried any DVCS?
Coming from Subversion?

Tried any DVCS?
Tried any DVCS?

Coming from Subversion?

Using Git today?
✓ Coming from Subversion?
✓ Tried any DVCS?
✓ Using Git today?
Certified Git User
145+ commands
18 used on a daily basis
<table>
<thead>
<tr>
<th>init</th>
<th>checkout</th>
<th>bisect</th>
</tr>
</thead>
<tbody>
<tr>
<td>log</td>
<td>revert</td>
<td>clone</td>
</tr>
<tr>
<td>rebase</td>
<td>pack</td>
<td>diff</td>
</tr>
<tr>
<td>merge</td>
<td>pull</td>
<td>grep</td>
</tr>
<tr>
<td>commit</td>
<td>push</td>
<td>tag</td>
</tr>
<tr>
<td>add</td>
<td>status</td>
<td>branch</td>
</tr>
</tbody>
</table>
Distributed Version Control

A new idea?
1997
code co-op
2001

arch
2003

monotone
2003

SVK
2005

bazaar
2005

mercurial
2005

git
Why?
Centralized version control systems

- Subversion
- Perforce
- SourceSafe
- ClearCase
- PVCS
- CVS
- RCS
- Folders

CVS
Subversion
Perforce
ClearCase
SourceSafe
PVCS
CVS
RCS
Folders
“hated it with a passion...
...no way to do CVS right.
-Linus Torvalds”
Centralized VCS uses DB sequential index.
Git uses SHA-1 hash.
40 hex characters
use as little of it as is unique
treeish
treeish = **shorthand** for hashes
HEAD\textsuperscript{^\textregistered}
--since="2 weeks ago"
Hashable Objects

- Blob
- Tree
- Commit
- Tag
commit
c67db

- tree: a10b3
- parent: nil
- author: Fird
- committer: Matthew
- message: Major refactoring of the web content.

blob
7e8b1

- tree: 7e8b1 web
- blob: 9ab16 index.html

blob
9ab16

- html>
  <body></body>
  </html>
Major refactoring of the Javascript rendering engine.

Minor update to HTML

New language translations
Distributed connectivity
developer A
Three Stage Thinking
Stash

Working

Repo

Index

git clone

Remote

git checkout
repo

Index

working

repo

git clone

git checkout

edit project files

boss interrupts

git stash

remote
- git clone
- git checkout
- edit project files
- boss interrupts
- git stash
- test bug on master
- working as designed
git clone

edit project files

boss interrupts

git stash

test bug on master

working as designed

git stash apply
repo
Index
Remote

Stash
Working
Repo

- git clone
- git checkout
- edit project files
- boss interrupts
- git stash
- test bug on master
- working as designed
- git stash apply
- git add
- git commit
git clone

git checkout

edit project files

boss interrupts

git stash

test bug on master

working as designed

git stash apply

git add

git commit

git push
Working Offline
More available connectivity
More available connectivity
More demand to work without connectivity
More demand to work without connectivity
Checkin
Checkin
Add
Branch
List change log
Grep history
Rewrite history
Stash
Merge
Label
Remove
practically everything but push

Offline Anything
Checkin  
Add  
Branch  
List change log  
Grep history  
Rewrite history  
Stash  
Merge  
Label  
Remove  
practically everything but push
Matthew: Cool!
Another commit to my latest OSS project using Git!
Matthew: Cool! Another *commit* to my latest OSS project using *Git*!

Madelaine: I like *subversion* much better...
Storage Mechanics
Typical SCMs use *deltas*
CVS / Subversion / darcs / Mercurial
Delta storage gets slower as the history of a file gets longer.
Git uses **Directed Acyclic Graph** storage (**DAG**)
cp  -r  srcfolder  srcfolder.prev
Copy of the **entire tree** per checkin
hardlinks to existing **identical** blobs
zlib deflates every blob at commit
zlib deflates entire repo
400MB Subversion repo
convert to a 70MB Git repo
Speed
http://whygititisbetterthanx.com
data from http://whygitisbetterthanx.com/#git-is-fast
10 to 100 times Faster
400MB Subversion repo
convert to a 70MB Git repo
Branch, Merge & Rebase
Branch
git branch newbranch
Standard Branch

a32

2e2

8b3

9f1

Branch

d19

e69

Master/Trunk/MainLatest
Merge
Merges lump branch changes into a new unified commit
git merge --no-commit <sourcebranch>
git merge --squash <sourcebranch>
git merge <sourcebranch1> <sourcebranch2>
Rebase
Not a merge
simulates team members **taking turns** working
(one person at a time)
git checkout myfeaturebranch

git rebase master
Master/Trunk/MainLatest
Rebasing

Branch

Master/Trunk/Main/Latest

a32

2e2

8b3

d19

e69
Master/Trunk/MainLatest

8b3 ← 2e2 ← a32

Branch
Master/Trunk/MainLatest

a32

2e2

8b3

d19'

e69'
Who's Got Git?
Get source

For information about current problems and fixes, see Known issues.

This document describes how to set up your local work environment, how to use Repo to get the Android files, and how to build the files on your machine.

Related reading:
- For an overview of the entire code-review and code-update process, see Workflow.
- For reference details about Repo, see Using Repo and Git.

What's in the source?

For a description of all the projects that make up the Android source code, see Project layout. To see snapshots and histories of the files available in the public Android repositories, visit the GitWeb web interface.

The source is approximately 2.1GB in size. You will need 6GB free to complete the build.

Setting up your machine

To build the Android source files, you will need to use Linux or Mac OS. Building under Windows is not currently supported.

Linux
Rails is moving from SVN to Git
Posted by David April 02, 2008 @ 09:39 PM

We’ve been preparing for Rails to move the official source repository from Subversion to Git for some time now and it seems that it’ll happen over the next week or so. The premiere will happen alongside the official launch of Github.

The move will also mean that we’re going to be switching ticket tracking to Lighthouse. So now both our repository and ticket tracking will be powered by Rails applications, which is a nice bonus treat.

When the move happens, we’ll freeze the existing Subversion repository and the Trac installation. Both will live on for a long time to come, but will be entirely deprecated. This means that your existing svn:externals will not break, but if you want the latest edge, you’ll have to get it from the new git repository.

So now is a great time to learn more about Git in anticipation of this move. I recommend starting with the Git for SVN’ers crash course.
Perl migrates to git versioning control system.

The Perl Foundation has announced at the website switching the version control systems to git. According to the announcement Perl 5 would allow the language development team to take advantage of git's extensive offline and distributed versioning support.

Git is open source and readily available to all Perl developers. Among other advantages, the announcement notes that git simplifies commits, producing fewer administrative overheads for integrating contributions. Git's change analysis tools are also singled out for praise.

The transformation from Perforce to git apparently took over a year. Sam Vilain of Catalyst IT "spent more than a year building custom tools to transform 21 years of Perl history into the first ever unified repository of every single change to Perl." The git repository incorporates historic snapshot releases and patch sets, which is frankly both cool and historically pleasing. Some of the patch sets were apparently recovered from old hard drives, notching up the geek satisfaction factor even more.
Git at Apache

This is an collection of read-only Git mirrors of Apache codebases. The mirrors are automatically updated and contain full version histories (including branches and tags) from the respective source trees in the official Subversion repository at Apache.

Note that these Git mirrors are missing Subversion features like svn:ignore, svn:relocate-style and svn:keywords settings and support for empty directories.

Please contact the infrastructure-dev@apache.org mailing list if you have comments or suggestions regarding this service or want to see another Apache codebase mirroring.

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
<th>Git Clone URL</th>
<th>HTTP Clone URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>abdera.git</td>
<td>Apache Abdera</td>
<td>git://git.apache.org/abdera.git</td>
<td><a href="http://git.apache.org/abdera.git">http://git.apache.org/abdera.git</a></td>
</tr>
<tr>
<td>activemq.git</td>
<td>Apache ActiveMQ</td>
<td>git://git.apache.org/activemq.git</td>
<td><a href="http://git.apache.org/activemq.git">http://git.apache.org/activemq.git</a></td>
</tr>
<tr>
<td>ant.git</td>
<td>Apache Ant</td>
<td>git://git.apache.org/ant.git</td>
<td><a href="http://git.apache.org/ant.git">http://git.apache.org/ant.git</a></td>
</tr>
<tr>
<td>apr.git</td>
<td>Apache Portable Runtime</td>
<td>git://git.apache.org/apr.git</td>
<td><a href="http://git.apache.org/apr.git">http://git.apache.org/apr.git</a></td>
</tr>
<tr>
<td>archiva.git</td>
<td>Apache Archiva</td>
<td>git://git.apache.org/archiva.git</td>
<td><a href="http://git.apache.org/archiva.git">http://git.apache.org/archiva.git</a></td>
</tr>
<tr>
<td>axis2-c.git</td>
<td>Apache Axis2/C</td>
<td>git://git.apache.org/axis2-c.git</td>
<td><a href="http://git.apache.org/axis2-c.git">http://git.apache.org/axis2-c.git</a></td>
</tr>
<tr>
<td>axis2-java.git</td>
<td>Apache Axis2/Java</td>
<td>git://git.apache.org/axis2-java.git</td>
<td><a href="http://git.apache.org/axis2-java.git">http://git.apache.org/axis2-java.git</a></td>
</tr>
<tr>
<td>batik.git</td>
<td>Apache Batik</td>
<td>git://git.apache.org/batik.git</td>
<td><a href="http://git.apache.org/batik.git">http://git.apache.org/batik.git</a></td>
</tr>
<tr>
<td>buildr.git</td>
<td>Apache Buildr</td>
<td>git://git.apache.org/buildr.git</td>
<td><a href="http://git.apache.org/buildr.git">http://git.apache.org/buildr.git</a></td>
</tr>
<tr>
<td>camel.git</td>
<td>Apache Camel</td>
<td>git://git.apache.org/camel.git</td>
<td><a href="http://git.apache.org/camel.git">http://git.apache.org/camel.git</a></td>
</tr>
<tr>
<td>cassandra.git</td>
<td>Apache Cassandra (incubating)</td>
<td>git://git.apache.org/cassandra.git</td>
<td><a href="http://git.apache.org/cassandra.git">http://git.apache.org/cassandra.git</a></td>
</tr>
<tr>
<td>cayenne.git</td>
<td>Apache Cayenne</td>
<td>git://git.apache.org/cayenne.git</td>
<td><a href="http://git.apache.org/cayenne.git">http://git.apache.org/cayenne.git</a></td>
</tr>
<tr>
<td>chainsaw.git</td>
<td>Apache Chainsaw</td>
<td>git://git.apache.org/chainsaw.git</td>
<td><a href="http://git.apache.org/chainsaw.git">http://git.apache.org/chainsaw.git</a></td>
</tr>
<tr>
<td>cocoon.git</td>
<td>Apache Cocoon</td>
<td>git://git.apache.org/cocoon.git</td>
<td><a href="http://git.apache.org/cocoon.git">http://git.apache.org/cocoon.git</a></td>
</tr>
<tr>
<td>commons-codec.git</td>
<td>Apache Commons Codec</td>
<td>git://git.apache.org/commons-codec.git</td>
<td><a href="http://git.apache.org/commons-codec.git">http://git.apache.org/commons-codec.git</a></td>
</tr>
<tr>
<td>commons-collections.git</td>
<td>Apache Commons Collections</td>
<td>git://git.apache.org/commons-collections.git</td>
<td><a href="http://git.apache.org/commons-collections.git">http://git.apache.org/commons-collections.git</a></td>
</tr>
</tbody>
</table>
Grails Moves To Git

March 25, 2009 by graemerocher | grails, git | 4 comments.

The Grails source control repository has been officially moved to Github. The SVN repository will continue to act as a read-only mirror of the Git repository.

Recent Comments:
arguably the **best feature** of git
use git in **stealth mode**
git svn clone http://unfurl.com/trunk

//Hack, hack, hack

```bash
git commit -a -m 'Some changes'
git commit -a -m 'More changes'
git commit -a -m 'Final changes'
```
$git svn dcommit
Committing to http://ambientideas.unfuddle.com/svn/ambientideas_demo ..

  A anewfile2.txt
Committed r7
  A anewfile2.txt
r7 = 7772d0025962d976bcec34151155244c5af91153 (git-svn)
No changes between current HEAD and refs/remotes/git-svn
Resetting to the latest refs/remotes/git-svn

$
Subversion

- First class compatibility.
- Round trip support.
- Git commits = svn commits.
effortlessly **convert full history** when ready to migrate the team
git svn clone http://unfurl.com/trunk
You now have a **full git clone** of the SVN repo
Power off your Subversion server
3 reasons
`git cherry-pick a5b2ee`

- Merge in just one commit.
Cherry Pick

- `git cherry-pick a5b2ee`
- Merge in just one commit.
```bash
git log -S "SomeText"
```

Search blob contents history without checkouts.
```bash
git log -S "SomeText"
```

Search blob contents history without checkouts.
- `git bisect run mvn test`
- Binary-search for bug via failed tests.
- Manual or automated modes.
- Ant, Maven, or shell script that can return 0 or 1-127
git bisect run mvn test

Binary-search for bug via failed tests.

Manual or automated modes.

Ant, Maven, or shell script that can return 0 or 1-127
Twitter
@matthewmccull

Blog
http://www.ambientideas.com/blog
sidebar has all my social media links

Email
matthewm@ambientideas.com

GitHub
http://github.com/matthewmccullough
Image Credits

- http://www.ambientideasphotography.com
- http://flickr.com/photos/lenore-m/2903856664/
- http://en.wikipedia.org/wiki/Git_(software)
- http://flickr.com/photos/karenhorton/1583513014/
- http://flickr.com/photos/mashdnart/2545782407/
- http://www.flickr.com/photos/knmurphy/2506896257/
- http://www.flickr.com/photos/albyspace/1022035568/
- http://flickr.com/photos/michaelhays/3070238360/
- http://flickr.com/photos/d_vdm/509996632/