

Revision Control for the Rest of Us *

* SysAdmins, scripters and anyone who is not a
full-time Real Programmer™

Updated: 2017-11-21

JP Vossen, CISSP
bashcookbook.com

http://www.jpdomain.org/public/Revision_Control_for_the_Rest_of_Us.pdf
http://www.jpdomain.org/public/Revision_Control_for_the_Rest_of_Us.odp

Agenda

- The “rest of us?”
- What is revision control?
- Why should I care?
- Hints
- Which tool should I use?
 - etckeeper
 - git
 - Bazaar
 - Mercurial
 - Meld
- What next?
- URLs, Wrap up and Q&A

The “rest of us?”

- Not all of us are “Real Programmers” working on large-scale, world-wide distributed teams
- Not everyone realizes how useful revision control is
- If you use a computer, you probably do something that revision control can help with
- If you are a system administrator or scripter, you should be using revision control
- If you are any kind of programmer, revision control should be like breathing, it's critical and I shouldn't have to tell you to do it

What is revision control?

- A way to travel back in time to see what changed, and when, and who, and **why**
- Similar to Mac's Time Machine
- AKA “version control,” “source control,” “code management” and more
- https://en.wikipedia.org/wiki/Revision_control
- <http://betterexplained.com/articles/a-visual-guide-to-version-control/>
- <http://www.catb.org/esr/writings/version-control/version-control.html>

Why should I care?

- If you work on a development team, it's just required; but you already know that and aren't the audience for this.
- Have you ever changed something then wished you could undo the change?
- Have you ever forgotten why you changed something? (Or who changed it?)
- Would a new *backup* every time anything changed, detailing why it changed, and with the ability to compare (diff) the change ever be useful?

Hints

- Most useful for plain text: code, wiki markup, configs
 - /etc/
- Not good for binaries: ODF, OOXML, ZIP, ISO, music, videos, etc.
- Some people put some or all of \$HOME into revision control
- Coolest thing ever:
<https://github.com/dspinellis/unix-history-repo>

Which tool should I use?

- This can easily turn into a religious war
- If you know a tool already, use that
- If \$WORK uses or recommends a tool, use that
- What is the goal? Maybe use *etckeeper*?
- Otherwise use:
 - Git
 - Bazaar
 - Mercurial
- Don't use these unless you really, really have to:
 - Subversion
 - CVS

etckeeper

- Stores */etc/* in git, mercurial, bzr or darcs
- In the Debian/Ubuntu repos
- In EPEL for Red Hat/CentOS etc.
- Hooks in to apt/yum and commits changes before and after operations as needed
- Auto-commits changes once a day as needed
- You can manually commit changes using ``etckeeper commit 'some message'``
- **Much** easier than trying to do it yourself!
- But local (inside */etc/.vcs*), so you need to back it up!

git

- Has unquestionably won the war--all your friends already jumped off the bridge
- Has <https://github.com/>
- Is less actively user hostile than it used to be
- But no “native” Windows CLI version! (Feature? Or Bug?)
- Uses hashes and dates instead of human-readable revision numbers
- Is made out of razor blades and chainsaws—blazingly fast, extremely powerful, but dangerous; you **can** hurt yourself with it!
- History is (**very**) malleable (and git considers this a feature, not a bug)
- The git “index” is different; none of the other common tools have this (see ``git commit -p`` but also ``hg commit -interactive``!)
- I personally would argue that it's not suitable for beginners or casual users... But... It's everywhere and used by everyone... And...github...
 - See the “easygit” wrapper: <https://people.gnome.org/~newren/eg/>

Git URLs

- URLs:
 - [https://en.wikipedia.org/wiki/Git_\(software\)](https://en.wikipedia.org/wiki/Git_(software))
 - <https://git-scm.com/>
 - <http://xkcd.com/1597/>
 - Book: <https://git-scm.com/book/en/v2>
 - Book: O'Reilly's 400 page *Version Control with Git*
 - <http://stevebennett.me/2012/02/24/10-things-i-hate-about-git/>
 - *Microsoft put Windows into Git:*
 - <https://blogs.msdn.microsoft.com/bharry/2017/05/24/the-largest-git-repo-on-the-planet/>
 - <https://tech.slashdot.org/story/17/05/24/2136212/windows-switch-to-git-almost-complete-8500-commits-and-1760-builds-each-day>
 - <https://apple.slashdot.org/story/17/11/17/2119235/microsoft-and-github-team-up-to-take-git-virtual-file-system-to-macos-linux>

git demo

```
git help | less
git config --global user.name "Joe McUser"
git config --global user.email "jmcuser@example.com"
git init myrepo
cd myrepo
echo -e "#! /bin/bash -\necho 'Hello World.'" > hello.sh
git status
git add hello.sh
git status
git commit -m 'First commit'
git status
echo "echo 'New line'" >> hello.sh
git status
git diff (you also need to know git diff --staged!)
gitk
git commit -m 'Added a new line' (fails, need -a or...)
git log
```



Bazaar

- Was Canonical's answer to git
- Has <https://launchpad.net/>
- Lost the war and was not-quite-dead...but:
- New life, *Breezy* (*brz*, alias *bzr*): <https://launchpad.net/brz>
 - And moving to Python3
 - In Debian testing since around 2017-04
 - Wish list for “interactive” <https://bugs.launchpad.net/brz/+bug/1733643>
- Extremely user friendly with awesome docs
- Uses incrementing integer revision numbers
- Cross-platform Python with several GUI tools
 - qbzr (Qt), loggerhead (web), others
- History is immutable
- Not as fast as git, but that almost never matters
- URLs:
 - [https://en.wikipedia.org/wiki/Bazaar_\(software\)](https://en.wikipedia.org/wiki/Bazaar_(software))
 - <http://wiki.bazaar.canonical.com/Documentation>
 - <http://wiki.bazaar.canonical.com/Workflows>
 - Book: <http://wiki.bazaar.canonical.com/BazaarBook>

Bazaar demo

```
bzr help
bzr init myrepo
cd myrepo
echo -e "#! /bin/bash -\necho 'Hello World.'" > hello.sh
bzr status
bzr add hello.sh
bzr st
bzr commit -m 'First commit'
bzr st
echo "echo 'New line'" >> hello.sh
bzr st
bzr diff
bzr qdiff
bzr commit -m 'Added a new line'
bzr log
bzr qllog
```

Mercurial

- Still lost to Git but more active than Bazaar/Breezy
 - Started at the same time as Git for the same reason
- Has Atlassian <https://bitbucket.org/> (for now?)
- Extremely user friendly with good docs
- Uses incrementing integer revision numbers + a hex ID
 - The hex ID is unique and consistent across all repo clones, the integer isn't
- Cross-platform Python with several GUI tools
 - Built in web server (`hg serve`` then <http://localhost:8000/>)
- History is immutable
- Has `hg commit -interactive`` built in!
- Not as fast as git, but that almost never matters
- URLs:
 - <https://en.wikipedia.org/wiki/Mercurial>
 - <https://www.mercurial-scm.org/>
 - <https://www.mercurial-scm.org/guide>
 - Book: <http://hgbook.red-bean.com/>

Mercurial demo

```
echo -e "[ui]\nusername = Joe McUser <jmcuser@example.com>" >> ~/.hgrc
```

```
hg help | less
```

```
hg init myrepo
```

```
cd myrepo
```

```
echo -e "#! /bin/bash -\necho 'Hello World.'" > hello.sh
```

```
hg status
```

```
hg add hello.sh
```

```
hg st
```

```
hg commit -m 'First commit'
```

```
hg st
```

```
echo "echo 'New line'" >> hello.sh
```

```
hg st
```

```
hg summary
```

```
hg diff
```

```
hg ci -m 'Added a new line'
```

```
hg log
```

Meld

- Meld is an awesome GUI diff/merge tool, not a revision control tool
 - 2 way and 3 way!
- But it understands revision control and works with it
- Works on Linux and Windows, I never got it working on Mac
- URLs:
 - <https://en.wikipedia.org/>
 - <http://meldmerge.org/>



What next?

- The demos only show the most basic, single user, local (not-distributed) uses. The whole point of these tools is that they are distributed!
 - Right now, you have no backup! The repo is inside your dir, see `.git`, `.bzd` or `.hg`.
- But... Where you go from here and how you use the tools is one of those “it depends” situations.
- Pick one and use it for something. Better yet, do something simple in all three and see which one works and thinks like you do.
 - Or read all the books.
- You can convert between them using the “fast-export” and “fast-import” tools.
 - Worst case, see ESR's <http://www.catb.org/esr/reposurgeon/>
- They all have books, docs, and blog posts, and religious wars. But watch out for old data! A lot of the old reasons to pick one over the other are fixed and gone.

URLs, Wrap-up and Q&A

- http://www.jpdomain.org/public/Revision_Control_for_the_Rest_of_Us.pdf
- http://www.jpdomain.org/public/Revision_Control_for_the_Rest_of_Us.odp
- https://en.wikipedia.org/wiki/Revision_control
- <http://betterexplained.com/articles/a-visual-guide-to-version-control/>
- <http://www.catb.org/esr/writings/version-control/version-control.html>
- Unix history: <https://github.com/dspinellis/unix-history-repo>
- Git
 - “easygit” wrapper: <https://people.gnome.org/~newren/eg/>
 - [https://en.wikipedia.org/wiki/Git_\(software\)](https://en.wikipedia.org/wiki/Git_(software))
 - <https://git-scm.com/>
 - <http://xkcd.com/1597/>
 - Book: <https://git-scm.com/book/en/v2>
 - <http://stevebennett.me/2012/02/24/10-things-i-hate-about-git/>
 - Microsoft put Windows into Git
 - <https://blogs.msdn.microsoft.com/bharry/2017/05/24/the-largest-git-repo-on-the-planet/>
 - <https://tech.slashdot.org/story/17/05/24/2136212/windows-switch-to-git-almost-complete-8500-commits-and-1760-builds-each-day>
- Bazaar & Breezy
 - [https://en.wikipedia.org/wiki/Bazaar_\(software\)](https://en.wikipedia.org/wiki/Bazaar_(software))
 - <https://launchpad.net/brz>
 - <http://wiki.bazaar.canonical.com/Documentation>
 - <http://wiki.bazaar.canonical.com/Workflows>
 - Book: <http://wiki.bazaar.canonical.com/BazaarBook>
- Mercurial
 - <https://en.wikipedia.org/wiki/Mercurial>
 - <https://www.mercurial-scm.org/>
 - <https://www.mercurial-scm.org/guide>
 - Book: <http://hgbook.red-bean.com/>
- Questions?
- I'm on the PLUG list... jp@jpsdomain.org