Bazaar

Jaroslav Jindrak
Bazaar

- Version control system
  - Similar to Git, Mercurial or Subversion
  - Supports both distributed and centralised paradigm
- Open source, supported by Canonical
- Main CVS tool used on Launchpad.net
Availability

- Windows
- Linux
  - RHEL, CentOS, Fedora, Ubuntu, Debian …
- Unix
  - Solaris, AIX, HP-UX, OS X
- BSD
  - FreeBSD, PC-BSD
Git & Bazaar interface (1)

- git whoami
- git init
- git clone
- git branch
- git commit
- git revert
- git push
- git fetch/pull
- git status
- git merge
- git log
Git & Bazaar interface (2)

- git whoami
- git init
- git clone
- git branch
- git commit
- git revert
- git push
- git fetch/pull
- git status
- git merge
- git log
How standards proliferate:
(See: A/C chargers, character encodings, instant messaging, etc.)

<table>
<thead>
<tr>
<th>Situation: There are 14 competing standards.</th>
</tr>
</thead>
<tbody>
<tr>
<td>14?! Ridiculous! We need to develop one universal standard that covers everyone's use cases. Yeah!</td>
</tr>
<tr>
<td>Situation: There are 15 competing standards.</td>
</tr>
</tbody>
</table>

(XKCD 927)
Getting started

• `bzr whoami <identity>`
  • e.g. `bzr whoami “Xi Tin-pching <xtp@gov.cn>”`
• `bzr init`
  • Creates a branch (in git creates a repository)
• `bzr init-repo`
  • Creates a shared repository
• `bzr branch/clone`
  • Copies meta and object data of a branch
Cloning branches

- `bzr branch/clone <to-clone> [<new-name>]`
  - Creates a copy of `<to-clone>`
  - Clone is aliased to branch
  - e.g.
    - `bzr branch http://proj.web.com/a-branch/`
    - `bzr branch ~/repo/a-branch`
    - `bzr branch lp:project-name/a-branch`
Branch URLs

- `file:///`
  - Local file, default
- `sfpt:///`
- `bzr:///`
  - Bazaar smart server
- `ftp:///`
- `http:///`
  - Read-only
- `lp:`
  - Launchpad, format `lp:~`
Tracking changes

- `bzr add <file-name>`
  - e.g. `bzr add main.c`
  - Starts tracking a file
    - The file remains tracked, no need to re-add on every change
- `bzr commit -m <message> [<<file-names>>]`
  - Commits changes of tracked files
  - Possible to specify which files (similar to using `git add`)
  - By default commits changes made to the whole repo, to commit only the current subtree use `bzr commit -m <message>`.
Push/Pull

- `bzr push <where>`
  - By default pushes to origin
  - Unlike Git, it is not possible to push from a copy with divergent changes
    - Argument --overwrite can be used to push our copy to the parent, overwriting any changes
- `bzr pull <where>`
  - By default pulls from origin
  - Unlike Git, it is not possible to pull from an origin with divergent changes
Merge

- `bzr merge`
  - Pulls latest changes from the parent
  - Creates `.BASE`, `.OTHER`, `.THIS` versions of conflicted files
  - Highlights conflicts in local files
- `bzr resolve <file>`
  - Marks a file as resolver and ready for commit
Checkouts

- Basicly branches, but on commit apply the changes to the parent first and then to the local copy
- Using bzr update instead of bzr pull
  - Heavyweight checkouts
    - A branch that basically does the above
  - Lightweight checkouts
    - Proxy to the parent, no data downloaded
      - Does not work offline
Others

- `bzr rervert -r <revision> [<file>]`
  - Bazaar marks versions by revision Ids
  - Reverts all changes done after `<revision>`
- `bzr diff [<file>]`
  - Same functionality as git
- `bzr log`
  - Lists revisions
- `bzr status`
  - Lists added/modified/deleted files yet to commit
Paradigms

- Bazaar supports two basic paradigms
  - Centralised
    - Like subversion
    - Checkout/update
  - Distributed
    - Like git or mercurial
    - Push/pull
    - bzr bind/unbind
    - Can be used to turn between local and remote branch
Q&A

¯\_(ツ)_/¯
THE END