# 

# Version Control Key Points

Mike Jackson, The Software Sustainability Institute

This work is licensed under the Creative Commons Attribution License

Copyright © The University of Edinburgh 2012

See <http://software-carpentry.org/license.html> for more information.



Derived from Chris Cannam’s original at, <https://code.soundsoftware.ac.uk/projects/easyhg/wiki/SC2012BootcampPlan>.

## Prerequisites

Mercurial, BitBucket.

## Introduction

Cover VersionControl.ppt, slides 1-2.

Use Mercurial command-line.

EasyMercurial GUI as a visually appealing alternative – once the concepts are understood.

## Create a repository directory and add a file

hg and Mercury.

hg init

hg status file.txt

“?” means repository does not know about it.

hg add file.txt

hg status file.txt

“A” means repository has scheduled it for addition but not yet added it.

hg commit –m “. . .” file.txt

abort: no username supplied message

/home/user/.hgrc file contains common settings.

[ui]

username = Boot Camp <bootcamp@gmail.com>

Commit message records “why” changes were made.

hg commit –m “. . .” file.txt

“made a change” messages are redundant.

hg status file.txt

“”means repository knows about it and it’s up-to-date.

.hgignore can record files to ignore e.g. ~ files, .o files, .class files etc.

syntax: glob

\*~

Add to repository.

hg add .hgignore

hg commit –m “. . .” .hgignore

## Log, status, difference and revert

hg log

hg status

“M” means file has been modified.

hg commit

hg diff

“-“means removed line, “+”means added line.

hg revert

.orig copy of discarded file.

hg update 0

hg update N

hg update

All commands so far apply to multiple files and directories too.

hg diff –r M –r N

Alternative to remembering version numbers.

hg tag MY\_JOURNAL\_CODE

hg update 0

hg update MY\_JOURNAL\_CODE

**Question: we can record our history of changes but we still have a problem, what is it?**

**Answer: no backups!**

## Working by yourself with backups

Manual copy via SSH or copy to DropBox, but revision control systems support remote access.

<http://bitbucket.org/>

Log in.

Click Create a repository.

Name field: Software Carpentry Boot Camp.

Check Access level: This is a private repository.

Select Repository type: Mercurial.

Click Create repository.

Click Get started.

Click I have code I want to import.

Copy “push” URL

hg push <https://user@bitbucket.org/user/software-carpentry-boot-camp>

warning: bitbucket.org certificate with fingerprint

Mercurial 1.7.3 Mercurial and SSL problem warnings. Either we use an --insecure flag at the command-line or edit .hgrc.

[hostfingerprints]

bitbucket.org = 24:9c:45:8b:9c:aa:ba:55:4e:01:6d:58:ff:e4:28:7d:2a:14:ae:3b

Commit commits changes locally, push pushes committed changes to a remote repository.

hg push <https://user@bitbucket.org/user/software-carpentry-boot-camp>

Click Source

Click Commits

rm –rf localrepository

hg clone <https://user@bitbucket.org/user/software-carpentry-boot-camp>

cd software-carpentry-boot-camp/

cd ..

hg clone <https://user@bitbucket.org/user/software-carpentry-boot-camp> another-clone

hg commit

hg push <https://user@bitbucket.org/user/software-carpentry-boot-camp>

cd ../software-carpentry-boot-camp

hg incoming <https://user@bitbucket.org/user/software-carpentry-boot-camp>

hg pull <https://user@bitbucket.org/user/software-carpentry-boot-camp>

hg history

hg diff

hg update

Push changes, pull changes, check for changes – everything needed for collaboration.

## Working with colleagues

Get attendees to pair up into Owner and Partner. Get partner for you too.

Owner:

Click BitBucket, cog icon.

Click Access Management.

Enter username of Partner.

Select Write permission.

Owner:

hg clone <https://owner@bitbucket.org/owner/software-carpentry-boot-camp>

Partner:

hg clone <https://partner@bitbucket.org/owner/software-carpentry-boot-camp>

First username is for login, second is repository owner’s repository location.

Partner: edit file and commit.

Owner: check incoming.

**Question: why are there no incoming changes?**

**Answer: because the changes are only in Partner’s local repository. They need to be pushed.**

Partner: push.

Owner: check incoming, pull, history, update.

Partner and Owner: edit file substantially, make sure you both change the same lines.

Partner: commit, push.

Owner: commit, push.

“push creates new remote heads” or “Push failed” warning.

Owner: pull, history,

hg merge

Merged file is marked up with Partner’s and Owner’s changes.

Options: keep Partner’s changes and discard Owner’s, keep Owner’s changes and discard Partner’s, manually resolve by editing the file.

Owner:

hg resolve -m

“-m” is “mark resolved” and not “message”.

Owner: commit and push

hg annotate fishstew.txt

BitBucket Commits page and tree of changes.

Share when completed units of work, small enough to be reviewed within an hour.

## Quickie practical

Create a new repository, SoftwareCarpentry.

Add shell files to this directory.

Push repository to BitBucket.

Throughout the rest of the boot-camp keep pushing directories and files there!

## Conclusion

Show MAUS’s Bazaar code, <https://code.launchpad.net/maus/+branches>, branches in Bazaar are analogous to clones in Mercurial.

Show EasyMercurial.

Cover VersionControl.ppt, slide 3 onwards.