Mercurial Usage

http://www.selenic.com/mercurial

Create

From scratch
hg init ~/projects/myproject

From existing data
cd ~/projects/myproject
hg init
hg add . ~ "bark"

From existing repo
hg clone ~/projects/myproject
~/workspace/myproject-xip

Browse

Files changed in working directory
hg status

Changes to tracked files
hg diff

History of changes
hg log

History of changes for file/directory
hg log FILE ... SUBDIR...

Who changed what, when in a file?
hg annotate FILE

What changed between R1 and R5
hg diff R1 R5

A specific file from a specific rev
hg cat REV myfile

How many branches in the repo
hg heads

Is the current branch merged from others?
hg parents

Concepts

REPOSITORY: where Mercurial stores the revisions
WORKING DIR: where you do your changes (work)
CHANGESET: a set of changes committed at once

REV: number that identifies a revision
REVISION: change set integrated into the repository
PARENT: immediate ancestor of a revision or the WD BRANCH: follow a given child from a revision
MERGE: a revision with two parents
HEAD: latest revision of a branch
TAG: name for a specific revision

PATCH: data to update files from R1 to R2
BUNDLE: patch with permissions and names support

Revert

Revert to parent rev of the working dir
hg revert

Content of a file to a previous rev
hg revert REV

Working directory content to a previous rev
hg revert -C

Current revision to a previous rev
hg update -C REV
(Rev will be new parent, local changes are cleaned)

Your last commit / pull / import / unbundle
hg rollback (you cannot undo a rollback)

Merge

Merge with a given revision
hg merge REV

(REV will be added as the second parent)

View which revisions you are merging
hg parents

Undo your messed-up merge
hg update -C tip
(works only if you merged from tip)

Update

Synchronize with main (or other) repository
hg pull OTHER_REPO

(but this does not change the working directory)

Current revision to latest one
hg update REV
(only works when on the same branch)

Apply a patch that some sent you
hg import PATCHFILE
hg unbundle BUNDLE

Publish

Publish your local changes to the "main"
hg push MAIN_REPO

(but you have to be in sync with "main")

Prepare a patch for other developers
hg export REVISIONS ...

= tagnumber.patch

Bundle changes to update other repo
hg bundle BUNDLEFILE hg OTHER_REPO

(see push when possible)

Mark a version / milestone
hg tag v1.0

Summary

Create
hg init [DIR] ... SOURCE [DIR]

Browse
hg status [OPTION] ... [FILE]
hg log [OPTION] ... [FILE]
hg annotate [REV] [F] [D] [E] [I] [X] [Q] [R] [V]
hg cat [OPTION] ... [FILE]
hg heads [R] [E] [V]

Revert
hg revert [REV] ... [PATTERN]

Update
hg update [REV] [DIR]
hg rollback [REV]

Publish
hg push [FILE] [REV] ...
hg export [FILE] [DIR]
hg annotate [REV]
hg tag [TAG] [FILE]

Options
hg help
hg version
hg clone
hg init
hg pull
hg revlist
hg log
hg annotate
hg cat
hg heads