

# Mercurial Usage

<http://www.selenic.com/mercurial>

## Create

### From scratch

```
hg init ~/projects/myproject
```

### From existing data

```
cd ~/projects/myproject
hg init
hg add -X "*.bak"
```

### From existing repo

```
hg clone ~/projects/myproject
~/workspace/myproject-wip
```

## 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 myfile
```

### What changed between R2 and R5

```
hg diff -r2 -r5 | diffstat
```

### A specific file from a specific rev

```
hg cat -rREV 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)  
**CHANGESSET** : a set of changes committed at once

**REV** : number that identifies a revision  
**REVISION** : changeset 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  
**TIP** : latest head  
**TAG** : name for a specific revision

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

## Revert

### Revert to parent rev of the working dir

```
hg revert
```

### Content of a file to a previous rev

```
hg revert -rREV FILE  
(but the working dir keeps the same parents)
```

### Working directory content to a previous rev

```
hg revert -rREV  
(but this does not change the current rev).
```

### 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 (you should have a clean WD)  
(REV will be added as the second parent)
```

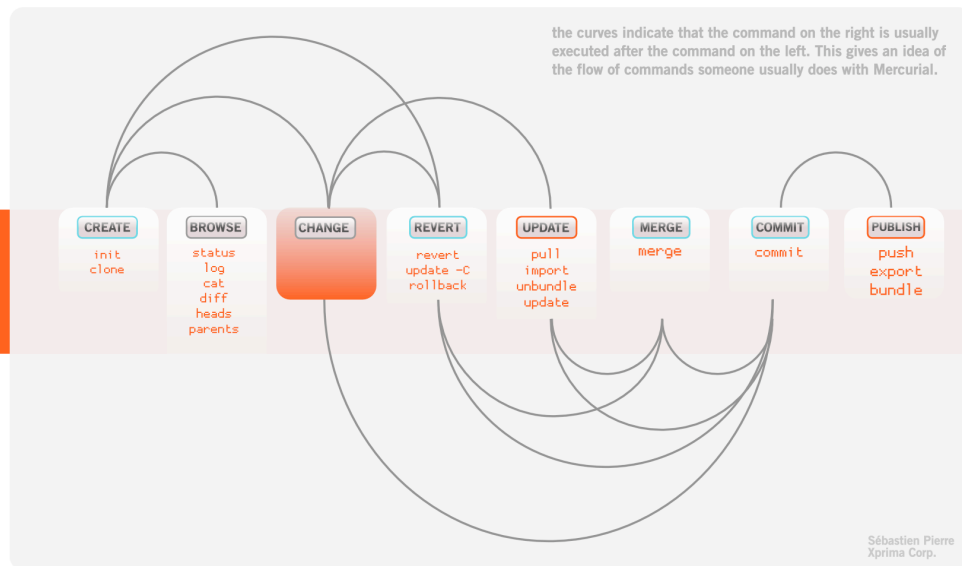
### View which revisions you are merging

```
hg parents
```

### Undo your messed-up merge

```
hg update -C tip  
(works only if you merged from tip)
```

## Commands Sequence



## 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... > mypatch.patch
```

### Bundle changes to update other repo

```
hg bundle BUNDLEFILE, hg OTHER_REPO  
(use push when possible)
```

### Mark a version / milestone

```
hg tag v1.0
```

## Summary

### Create

```
hg init [DEST]  
hg clone [OPTION]... SOURCE [DEST]
```

### Browse

```
hg status [OPTION]... [FILE]...  
hg diff [-a] [-I] [-X] [-r REV1] [-r REV2] [FILE]...  
hg log [OPTION]... [FILE]...  
hg annotate [-r REV] [-a] [-u] [-d] [-n] [-c] FILE...  
hg cat [OPTION]... FILE...  
hg heads [-b] [-r <rev>]  
hg parents [-b] [REV]
```

### Revert

```
hg revert [-I PATTERN] [-X PATTERN] [-r REV] [NAME]...  
hg update [-b TAG] [-m] [-C] [-F] [REV]  
hg rollback
```

### Update / Merge

```
hg pull [-u] [-e FILE] [-r REV]... [--remotecmd FILE] [SOURCE]  
hg merge [-b TAG] [-f] [REV]  
hg import [-p NUM] [-b BASE] [-F] PATCH...
```

### Publish

```
hg push [-f] [-e FILE] [-r REV]... [--remotecmd FILE] [DEST]  
hg export [-a] [-o OUTFILESPEC] REV...  
hg bundle FILE DEST  
hg tag [-l] [-m TEXT] [-d DATE] [-u USER] [-r REV] NAME
```