

Mercurial Usage

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

始める

ゼロから

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

既存のデータから

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

既存のリポジトリから

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

見る

作業ディレクトリの変更ファイル

```
hg status
```

管理下のファイルの変更点

```
hg diff
```

変更履歴

```
hg log
```

指定ファイル・ディレクトリの変更履歴

```
hg log FILE ... SUBDIR/ ..
```

誰がいつどこを変えた?

```
hg annotate myfile
```

Rev.2からRev.5までの変更点

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

指定リビジョンのファイル

```
hg cat -rREV myfile
```

リポジトリに存在するブランチ

```
hg heads
```

現在のブランチを他とマージしたのか?

```
hg parents
```

概念

リポジトリ: Mercurial がここにリビジョンを保存
作業ディレクトリ: 変更はここで行う
チェンジセット: 一度にコミットした一連の変更

REV: リビジョンの識別番号
リビジョン: リポジトリに取り込まれたチェンジセット
親: リビジョンまたは作業ディレクトリの直近の祖先
ブランチ: あるリビジョンから特定の子まで
マージ: 親が2つあるリビジョン
ヘッド: ブランチの最新のリビジョン
TIP: 最新のヘッド
タグ: 特定のリビジョンにつけた名前

パッチ: ファイルをr1からr2へ更新するデータ
バンドル: パーミッションとリネームを扱えるパッチ

取り消す

作業ディレクトリを親リビジョンへ

```
hg revert
```

ファイルの内容を以前のリビジョンへ

```
hg revert -rREV FILE  
(ただし、作業ディレクトリの親リビジョンは変わらない)
```

作業ディレクトリの **内容** を以前のリビジョンへ

```
hg revert -rREV  
(ただし、現在のリビジョンは変わらない)
```

以前のリビジョンへ

```
hg update -C REV  
(REV が新しい親になり、ローカルの変更は取り除かれる)
```

直近の **commit / pull / import / unbundle** を

```
hg rollback ⚠️ (rollback は元に戻せません!)
```

マージ

特定のリビジョンとマージ

```
hg merge REV (きれいな作業ディレクトリで)  
(REV が第2の親になります)
```

どのリビジョンとマージするのかチェック

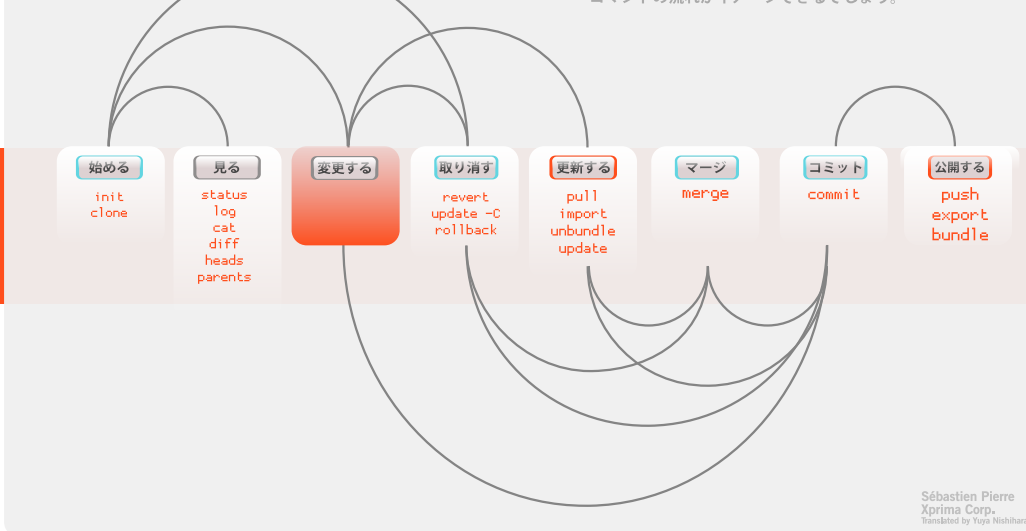
```
hg parents
```

ぐちゃぐちゃになったマージを取り消す

```
hg update -C tip  
(tip からマージした場合のみ)
```

コマンドの流れ

曲線の左側のコマンドを実行した後、右側のコマンドを使うことがよくあります。これを見れば Mercurial を使う上でのコマンドの流れがイメージできるでしょう。



Sébastien Pierre
Xprima Corp.
Translated by Yuji Nishitani

更新する

メイン(か他の)リポジトリと同期

```
hg pull OTHER_REPO  
(ただし、作業ディレクトリは更新しません)
```

現 **リビジョン** を最新のものに

```
hg update REV (同じブランチの場合のみ)
```

受け取ったパッチを適用

```
hg import PATCHFILE  
hg unbundle BUNDLE
```

公開する

ローカルの変更を「メイン」へ公開

```
hg push MAIN_REPO  
(その前に「メイン」と同期している必要あり)
```

他の開発者へ渡すパッチを作る

```
hg export REVISIONS... > mypatch.patch
```

他のリポジトリの更新用に変更をまとめる

```
hg bundle BUNDLEFILE.hg OTHER_REPO  
(できるだけ push を使ってください)
```

バージョン・マイルストーンに印をつける

```
hg tag v1.0
```

要点

始める

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

見る

```
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]
```

取り消す

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

更新・マージする

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

公開する

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