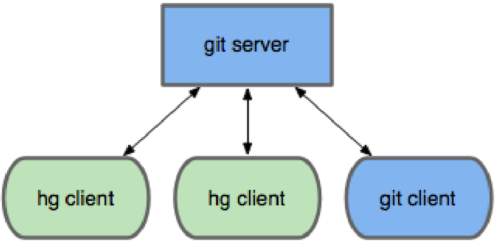
# the Hg-Git mercurial plugin

This is the Hg-Git plugin for Mercurial, adding the ability to **push to and pull from a Git server** repository from Mercurial. This means you can collaborate on Git based projects from Mercurial, or use a Git server as a collaboration point for a team with developers using both Git and Mercurial.

The Big Idea

The Hg-Git plugin can **convert commits/changesets losslessly** from one system to another, so you can push via a Mercurial repository and another Mercurial client can pull it. In theory, the changeset IDs should not change, although this may not hold true for complex histories.



Commands

## Clone a Git repository

```
hg clone git://github.com/schacon/some-repo.git
```

Like with normal `hg clone`, you can optionally specify a destination directory.

## Push an existing Hg repository to Git

```
$ cd mercurial-repo
$ hg bookmark -r default master # so a ref gets created
```

To avoid specifying the repo path when you push and pull, edit `.hg/hgrc` and add:

```
[paths]
default = git+ssh://git@github.com/schacon/some-repo.git
```

See **the Mercurial docs**for more detail on path settings.

```
$ hg push
```

This will convert all Mercurial data into Git objects and push them up to the Git server.

```
$ hg pull
```

That will pull down any commits that have been pushed to the server in the meantime and give you a new head that you can merge in.

Installing

## Installing Hg-Git the easy way:

First, make sure that you have a working C compiler on your system. On Windows, one option is the free **Microsoft Visual C++**. (Be sure to install the 2008 version for compatibility reasons.) On Debian-style Linux, run `apt-get install python-dev`.

Secondly, run `easy_install hg-git`. If you don't have easy_install available, you can get it as part of Python's **setuptools**package.

Alternatively, on Windows, **TortoiseHg**comes with hg-git (and Dulwich), though it **still needs to be enabled**as shown below (or through the TortoiseHg settings).

Lastly, make sure the following is in your `~/.hgrc`:

```
[extensions]
hgext.bookmarks =
hggit =
```

...and that's it!

## Installing Hg-Git manually

Use this method if you'd like to install a clone of the Hg-Git repository rather than a static package, or if you'd like to install Hg-Git in a location you choose yourself.

First, install version 0.8.0 or newer of **Dulwich**. You can do `easy_install 'dulwich>=0.8.0'` if you have **setuptools**installed. Next, clone **the Hg-Git repository**somewhere. Lastly, make the 'extensions' section in your '`~/.hgrc`' file look something like this:

```
[extensions]
hgext.bookmarks =
hggit = [path-to]/hg-git/hggit
```

That will enable the Hg-Git extension for you. The bookmarks section is only required prior to hg 1.7. Bookmarks will be translated to git heads when pushing.

Dependencies

Different versions of hg-git are known to work with different versions of Mercurial. For the most accurate info, go to **the source on bitbucket**, click the dropdown menu that says "default", click the "Tags" tab, select the version you've installed, and click on the file "Makefile". There's a line that starts with "all-version-tests" which lists the versions of Mercurial known to work.

It also needs Dulwich version 0.8.0 or later; Dulwich is a pure-Python implementation of the Git file formats and protocols. There are **no Git binary dependencies**: you do not need to have Git installed on your system.