

Git/Hg Rosetta Stone

hg	git
hg cat -r rev some_file	git show rev:some_file
hg clone http://hg.sympy.org/sympy-git.hg	git clone git://git.sympy.org/sympy.git
hg clone -U http://hg.sympy.org/sympy-git.hg	git clone --bare git://git.sympy.org/sympy.git
hg diff	git diff HEAD
hg status	git status
hg status -c	git ls-files -t grep '^H'
hg manifest	git ls-tree -r --name-only --full-tree HEAD
hg parents	git show --pretty=format:'%P' -s
hg commit	git commit -a
hg record	git add -p; git commit # or, for a more detailed interface: git add -i; git commit
hg email -r tip	git send-email HEAD^ # or: git format-patch HEAD^ ; git send-email 0001-Add-whitespace.patch
hg view	gitk, git gui
hg help command	git help command
~/hgrc	~/gitconfig
.hg/hgrc	.git/config
hg paths	git remote -v
editing paths in .hg/hgrc	git remote add name url # see "git help remote" for more info how to edit paths; alternatively, you can edit them by hand in .git/config too
.hgignore	.gitignore
.hg/hgrc [ui] ignore	.git/info/exclude
hg add	git add (note, it adds <code>_content_</code> to index; can work on a hunk-by-hunk basis with <code>-p</code>)
hg rm	git rm
hg push	git push
hg pull	git fetch
hg pull -u	git pull
hg addremove	git add -A (or: git add .; git ls-files --deleted xargs git rm)
hg revert -a	git reset --hard
hg revert some_file	git checkout some_file
hg purge	git clean -fd
hg purge --all	git clean -fdx
hg strip 2fccd4c	git reset --hard 2fccd4c^ (on a normal repository) git reset --soft 2fccd4c^ (on a bare repository)
hg export	git format-patch
hg import --no-commit some.patch	git apply some.patch
hg import some.patch	git am some.patch
hg out	git fetch && git log origin/master..
hg in	git fetch && git log ..origin/master
hg update tip	git checkout HEAD # or this: "git checkout master", or "git merge FETCH_HEAD", depending on what you did before this
hg update -C	git checkout -f
hg update some.branch	git checkout some.branch # Note that "git branch some.branch" is not the same as this.
hg up --date 2014-01-01	git checkout `git rev-list -n 1 --before="2014-01-01" master`
hg qimport	stg something (A separate patch manager extension is probably not necessary in git -- normal workflow combined with git rebase -i should cover your needs)
hg qpush	(see hg qimport)
hg qpop	(see hg qimport)
hg qimport -r tip	?
hg qnew -f some.patch	?
hg resolve -a -m	git add -u
hg root	git rev-parse --show-toplevel
hg glog	git log --graph --all --decorate # or: git log --graph --all;
hg verify	git fsck
hg branches	git branch -a
hg branch	git rev-parse --abbrev-ref HEAD
hg rollback	git reset HEAD~