Introduction to Sage Development

Mike Hansen

May 4, 2010

Introduction to Sage Development

Introduction to Sage Development

There are many ways to contribute to Sage, but I'm going to specifcally changing code in the Sage library and making it availble for others or submitting it for inclusion in Sage.

Trac

All of Sage development is done via the *Trac server* which can be found at *http://trac.sagemath.org*.

Mercurial

Sage uses the program Mercurial (*hg*) to manage all of its source code.

Mercurial

Sage uses the program Mercurial (*hg*) to manage all of its source code.
Mercurial stores all

Mercurial Queues

An extension to Mercurial that allows one to easily work with collections of patches.

Mercurial Queues

- *hg qnew*: Create a new patch
- *hg qpop*: Move a patch from the applied stack to the unapplied one.
- *hg qpush*: Move a patch from the unapplied stack to the applied one.
- *hg qrefresh*: Update the current patch to reflect the working directory
- *hg qtop*: Show the current patch
- *hg qapplied*: Print the applied stack.
- *hg qunapplied*: Print the unapplied stack
- *hg qseries*: Print all of the patches in order

/.hgrc

We need to enable Mercurial Queues by editing the .hgrc file.

```
[ui]
username = Mike Hansen <mhansen@gmail.com>

[extensions]
hgext.mq=
```

Typical Workflow

```
Run hg qinit if necessary

Make changes to the code.

hg qnew -f name_of_patch.patch

hg export name_of_patch.patch > /path/to/file.patch

Upload the patch to trac.
```