Working with Mercurial in the JMC Project

This page contains some tips and tricks for working with Mercurial in the JMC Project for Committers.

Getting Started

1. First install Mercurial (different on different platforms)
2. Next ensure that you have an OpenJDK user (http://openjdk.java.net/census)
   If you need an OpenJDK user name and/or generate a key, follow instructions here: http://openjdk.java.net/guide/producingChangeset.html#jdkUserName
3. Next install the defpath plug-in (http://openjdk.java.net/projects/code-tools/defpath/)
4. Next create a ~/.hgrc file and fill it out according to the example below.
5. Next clone the JMC source:

   ```
   hg clone http://hg.openjdk.java.net/jmc/jmc
   ```

Example minimal .hgrc file:

```
[ui]
username = youropenjdkname <your.mailaddress@example.com>

[extensions]
defpath = /usr/local/bin/hgext/defpath.py

[defpath]
username = youropenjdkname
```

Example, more comprehensive, .hgrc file:

```
[ui]
username = youropenjdkname <your.mailaddress@example>
merge = p4merge
editor = nano

[extensions]
defpath = /usr/local/bin/hgext/defpath.py
hgext.extdiff =

[extdiff]
cmd.p4diff = /Applications/p4merge.app/Contents/MacOS/p4merge

[tortoisehg]
vdiff = p4diff

[merge-tools]
p4merge.priority = 100
p4merge.premerge = True
p4merge.gui = True
p4merge.args = $base $local $other $output
p4merge.diffargs = $parent $child
p4merge.diff3args = $child $parent1 $parent2
p4merge.dirdiff = False
p4merge.checkchanged = True

[defpath]
username = youropenjdkname

[committemplate]
changeset = JMC-: <synopsis-of-symptom>\nSummary: <summary-of-codechange>\nReviewed-by: <reviewer>\n{desc}\n\n
    HG: Enter commit message.  Lines beginning with 'HG:' are removed.
    HG: {extramsg}
    HG: --
    HG: user: {author}\n{ifeq(p2rev, "-1", "",
   "HG: branch merge\n")
    }HG: branch '{branch}'\n{if(activebookmark,
   "HG: bookmark '{activebookmark}'\n")   }{subrepos %
   "HG: subrepo {subrepo}\n"               }{file_adds %
   "HG: added {file}\n"                    }{file_mods %
   "HG: changed {file}\n"                  }{file_dels %
   "HG: removed {file}\n"                  }{if(files, "",
   "HG: no files changed\n")}
```

You may also want to set up so that you can use p4merge for merging and diffing, and also use a template for commits.

If the change comes from a non-Author, don't forget to set "Contributed-by:".

To diff using p4merge, simply use the command:

```
 hg p4diff
```

To initialise defpath, after installed and configured, run:

```
 hg defpath -d
```

Review

Send either a unified diff or a webrev to the jmc-dev@openjdk.java.net mailing list

Webrev short hand:

ksh /usr/local/bin/webrev.ksh generates the webrev

rsync -avz <folder (named as the bugid) containing the rev> <openjdkid>@cr.openjdk.java.net:~/

Getting and applying a patch from a webrev:

Update your local clone
cd JMC_ROOT
hg pull
hg update

Download the patch

cd <some folder>
wget http://cr.openjdk.java.net/~hirt/JMC-5922/webrev.1/jmc.patch

Apply the patch

cd JMC_ROOT
hg import --no-commit <some folder>/jmc.patch