How to generate a patch file

(From : http://mercurial.selenic.com/wiki/CommunicatingChanges)

import/export

This method is useful for receiving small numbers of changes from contributors and is a traditional method for open source projects. This is primarily used in scenarios where changes get reviewed before being applied to a central repository.

In combination with the push/pull method, it's common to post patches generated by export to a mailing list, along with instructions for pulling from their source repository.

This can also be useful for "cherry-picking" individual patches from one repo into another.

```
$ hg export 1135
# HG changeset patch
# User Thomas Arendsen Hein <thomas@intevation.de>
# Node ID e455d91f62598b8f255ce6c0291afe8f8565e0d2
# Parent  2cd33ea2f66bae0eb7415cfcd7eab88566fdb1aa
Variable 'body' was missing in patchbomb script.

diff -r 2cd33ea2f66b -r e455d91f6259 contrib/patchbomb
--- a/contrib/patchbomb Sun Aug 28 16:30:40 2005
+++ b/contrib/patchbomb Sun Aug 28 16:52:55 2005
@@ -107,6 +107,7 @@
     def makepatch(patch, idx, total):
         desc = []
         node = None
+        body = ''
         for line in patch:
             if line.startswith('#'):
                 if line.startswith('# Node ID'): node = line.split()[-1]
$ hg export 1135 > ../body.patch
$ cd ../work
$ hg import ../body.patch
```

bundle/unbundle

This method allows communication of patches by exchanging "bundles": a compressed group of changesets in a native file format. These bundle files can then be exchanged via email attachments, FTP, floppy disk, etc.

This also allows you to publish your changes without publishing a copy of the entire project history.

```
$ hg bundle changes.hg http://upstream/repo
searching for changes
$ scp changes.hg server:public_html
$ cd ../other
$ hg unbundle http://server/~user/changes.hg
adding changesets
adding manifests
adding file changes
added 13 changesets with 20 changes to 6 files
```