Last revised: 2012-Jan-11

Using Mercurial for Computational Physics Assignments

Here we step you through creating a private repository on the code hosting site Bitbucket, which you can then share with the 780.20 instructors. You will use Mercurial to work on local versions of the files for problem sets and your project, then "push" a copy to Bitbucket.

Setting up a Bitbucket repository

1. First you need to establish both your username and the default editor in your `$HOME` directory (give the command `echo $HOME` to find out where this is). Use your editor to create the configuration file `.hgrc` (e.g., `nedit $HOME/.hgrc`, note the period at the beginning) in the following format, substituting your name, email, and favorite editor:

   ```
   [ui]
   username = Dick Furnstahl <furnstahl.1@osu.edu>
   editor = nedit
   ```

   (Use `hg help configure` to get more details about this file. Note that the email address has `<>`'s around it.) If you end up having trouble with `$HOME` (e.g., Mercurial doesn't find the username or editor), you can instead name the file `hgrc`, with no period, and put it in the `.hg` subdirectory that you create locally with an `hg clone` command (in the **Submitting homework assignments** section below).

2. Go to `http://bitbucket.org` and click on "Sign up now, free". This should take you to a page that says something like "Sign up for a free 5 user account".

3. Fill out the form with the email address from step 1 and your choice of username (no spaces, but something like `Dick_Furnstahl` is ok) and password.

4. When you have established your account, click on "create repository" under the "Respositories" menu at the top.

5. You'll get a form where you can specify the Name and Description of a new repository. Call your repository `780_LASTNAME_FIRSTNAME`, e.g., `780_furnstahl_dick`. In Description, put something like "Repository for 780.20 Computational Physics assignments". Check the box marked "private" (we don't want this to be readable by the public!) and leave "issue tracking" and Wiki off. You don't need to fill in the Website or choose a language.

6. Click on "create repository" and you are in! You'll get an email from bitbucket.org to confirm your email address.

7. Now give the instructors access to your repository. If you are not already there, go to your 780 repository by selecting it from the pull-down "Repositories" menu along the top. You should get to a page with tabs ("Overview", "Downloads", etc.). Select the "Admin" tab. On the left, select "Access Management" fill in `furnstahl` in the box under "Users", then click "Write". Repeat for `chrisorban`.

Submitting homework assignments

1. Create a 780.20 assignment directory on your local computer (which could be a laptop or one of the Physics Department computers) and "clone" your Bitbucket repository:

   ```
   hg clone https://MY_USER@bitbucket.org/MY_USER/MY_REPO
   ```

   (substituting for `MY_USER` and `MY_REPO` your username and the name you gave the repository!). This command can be cut-and-pasted from your bitbucket repository page. The website seems to need your names in lowercase for this clone command; if you used uppercase to create the respository, try changing the case in this command. This will create the directory `MY_REPO` (e.g., `780_furnstahl_dick`) on your local computer and an `.hg` subdirectory within it (view with `ls -a`). You should only need to clone your repository once, unless you change computers or locations on your disk (but it won't hurt to do it again).

2. Work on your assignment locally by creating subdirectories of `MY_REPO` (e.g., `PS1` for the first problem set or `Project` for your project) and creating and editing files there or copying files into them. Then update the files as decribed in the Session 2 Mercurial tutorial while in `MY_REPO`. For example:

   - to add files to the repository list, use `hg add`
   - to delete files already in the repository, use `hg remove`
   - to check on all your changes, use `hg log`

   and so on.

3. Commit your changes regularly to the local repository (every time you've made significant changes or when you end a session):

   ```
   hg commit -m "Initial commit of all files to the repository"
   ```

   if you want to include a comment in one step, or use `hg commit` and your editor will pop up a window where you type a comment to document the changes you made.

4. When you are ready to update the repository on Bitbucket (which you can do at any time and repeatedly), you "push" your changes using:

   ```
   hg push https://MY_USER@bitbucket.org/MY_USER/MY_REPO
   ```

   (the address is the same you used for the `hg clone` command). This will make a snapshot copy that the instructors can access.