Version Control through Bitbucket

In this tutorial, we look at how to use the bitbucket version-control system.

Why use a version control system? When developing software, we typically go through many versions. Each day we may add some new code, and sometimes we will rewrite older code. The first advantage that a version control system gives us is the ability to return to older versions of our code, to try to work out where a particular error occurred, perhaps. When you use a version- control system with an online repository, like bitbucket, you also have the advantages of automatic backups and availability of the latest version of your code from anywhere. No more worrying whether the code on your memory stick is more up-to-date than the code on your home computer… A final advantage, which we won't be using here, is that a version-control system makes it easier to work in teams on the same software - each programmer can work on their own "branch" of the code, which can then be "merged" into the main version when it has been checked for correctness.

There are many version-control systems, and many related repositories. In this module we will be using git as the version-control system, and storing our repositories on bitbucket. Alternatives to git include cvs, svn, mercurial and bazaar. You can read about them at http://www.smashingmagazine.com/2008/09/18/the-top-7-open-source-version-control-systems/. Alternatives to bitbucket include google code, github, sourceforge and launchpad. We use bitbucket because it is free to create a private repository - just what you need to store your coursework.

Create a bitbucket account by going to the website: https://bitbucket.org/ Email your username to your tutor. Your tutors' email is h.cheng2@herts.ac.uk and his bitbucket username is: huicheng.

The following sections demonstrate how to use version-control with Netbeans. You should be familiar with the usual ways of working with Netbeans before attempting to use version control.

1. Creating a New Project on Bitbucket

The project will be hosted simultaneously on your own computer and on bitbucket, meaning that sourcecode will be kept simultaneously on your own computer, and on the bitbucket site. You first need to create a project on bitbucket.
Go to your bitbucket homepage, which is `http://bitbucket.org/USERNAME` , for example, the homepage is: http://bitbucket.org/uhfoodtutor (This uhfoodtutor is just an example, it could be any username). Under the "Repositories" menu select "Create repository" (the big + sign on the left)
You will then be asked to enter the repository details. You **only need to** enter the name (which should not have any spaces, for example, example1) and ensure the "git" radio button is selected. Finally click on the "Create repository" button.
The screen will then show "Let's put some bits in your bucket" and you will see the http link like:

```
https://uhfoodtutor@bitbucket.org/uhfoodtutor/example1.git
```

Notice how the URL includes your username (uhfoodtutor) and project name (example1).

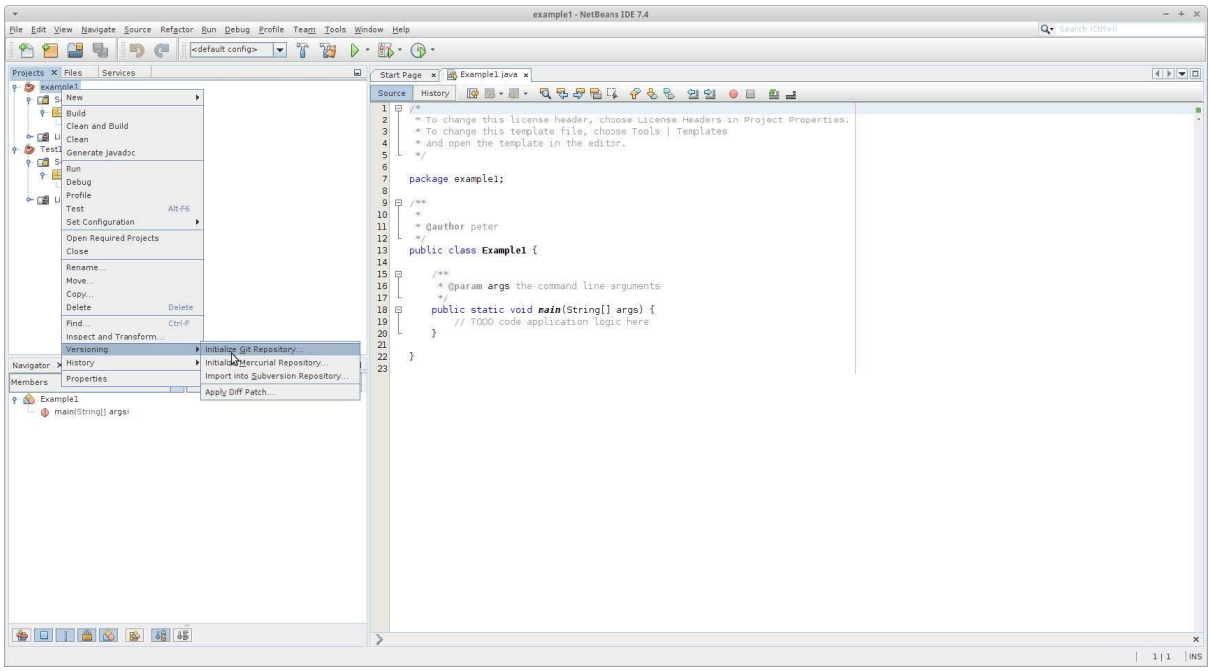
The `https` part is the url for your project, which we need to enter into Netbeans so Netbeans knows where to upload your code to. Make a note of this url, or copy it into the clipboard.

2. Develop-Commit-Push Cycle

We develop our Java code in Netbeans, and the use of version-control makes no difference to the usual pattern of development. However, once we have developed a piece of functionality, or simply finish for the night, we should "commit" and "push" the code.
A "commit" stores a version of the code at that moment along with a message, explaining what has just changed in the code. You should give the commit message some useful information, such as the part of the assignment you have completed.
A "push" places any committed versions of the code onto the remote repository. You do not need to push the code immediately after a commit.
If you use different computers to work on the same project, you will also need to "pull" the updated code down from the repository each time you change machine.
The use of version-control leads to a cycle in development, where the programmer develops some functionality (implementing and testing), commits the changes, and then pushes them to the repository.

2.1. Initialise Git

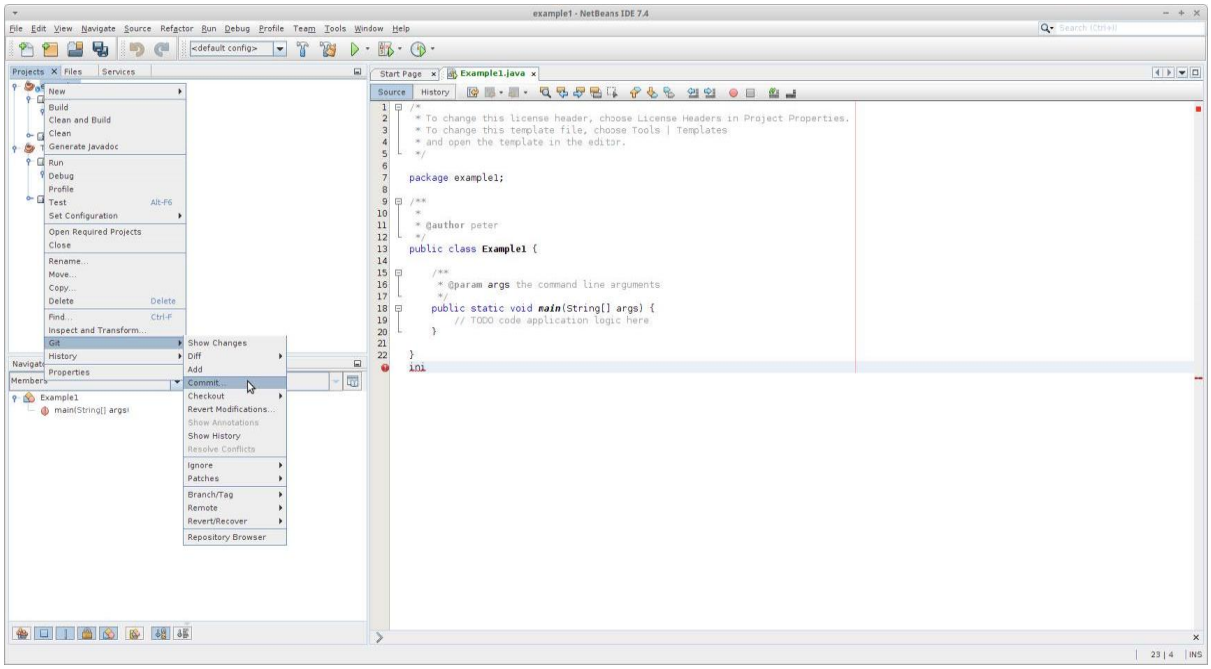
Before starting, we need to tell Netbeans that our project is using Git. To do this, right-click on the project name in Netbeans and select "Versioning" from the drop-down menu, and then "Initialize git repository . . . " from the submenu.



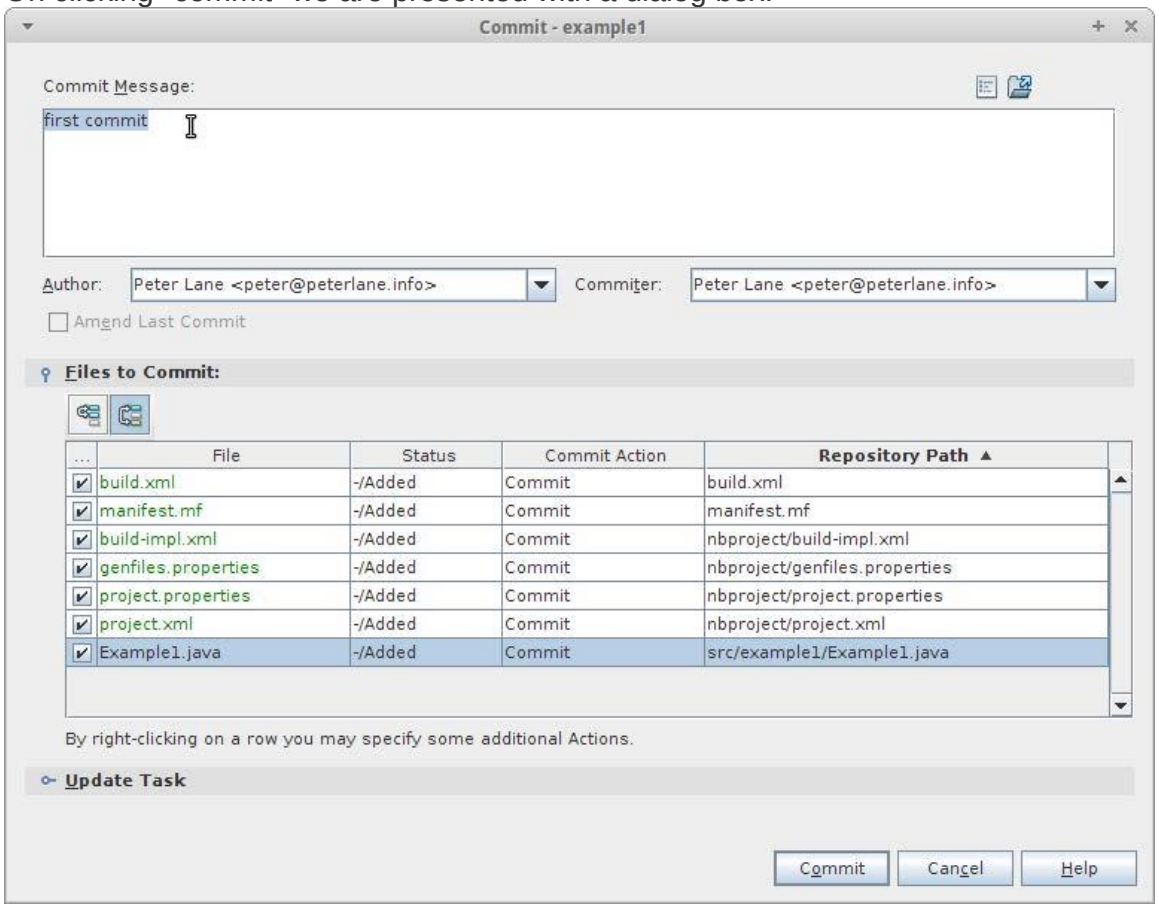
By default, the information for git will be stored in the project directory, so just click "OK". You will now find the "Versioning" menu has been replaced by one for "Git", and we need two options from that submenu, as explained next.

2.2. Commit

The "commit" operation is found in the drop-down menu obtained by right-clicking on the project name, selecting "git" and then "Commit":



On clicking "commit" we are presented with a dialog box:

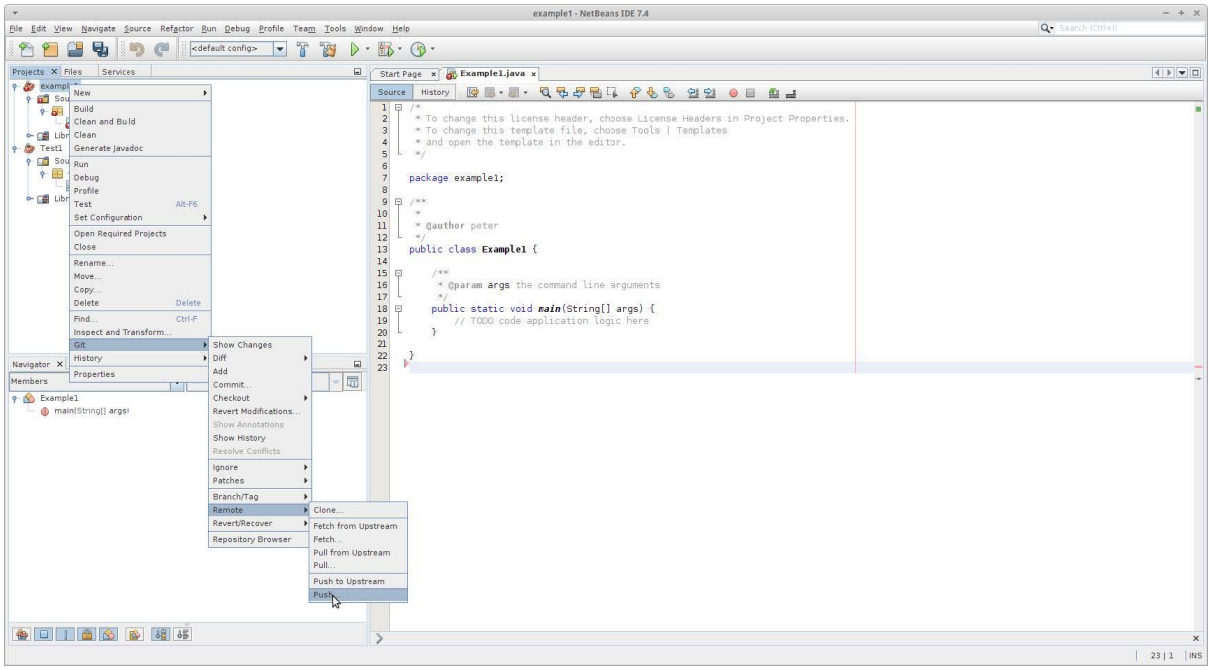


In the box labelled "commit message" enter some meaningful text about the commit. For example, it might be "completed exercise 1.3", or "finished tests for xxxxxx method".

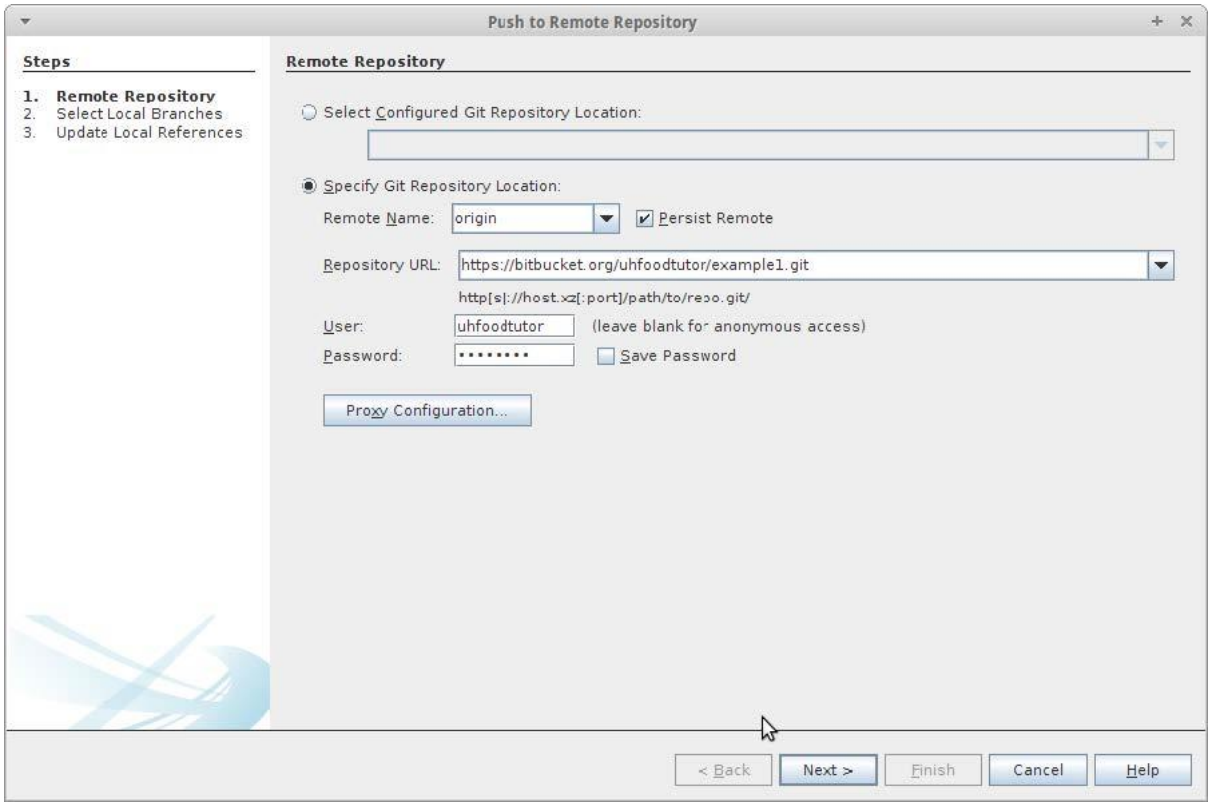
The central part of the dialog shows the files which will be stored within the commit. When happy with your message, click on "commit". Netbeans will then use git to store the current version of your project onto your computer. You can continue to develop, and make commits, which will be preserved on your machine. However, to save these commits securely, you need to push them to bitbucket.

2.3. Push

Once you have made some commits, you will want to upload your changes to the repository. This is known as a "push".
The "push" operation is found by right-clicking on the project name, selecting "Git", "Remote" and finally "push":

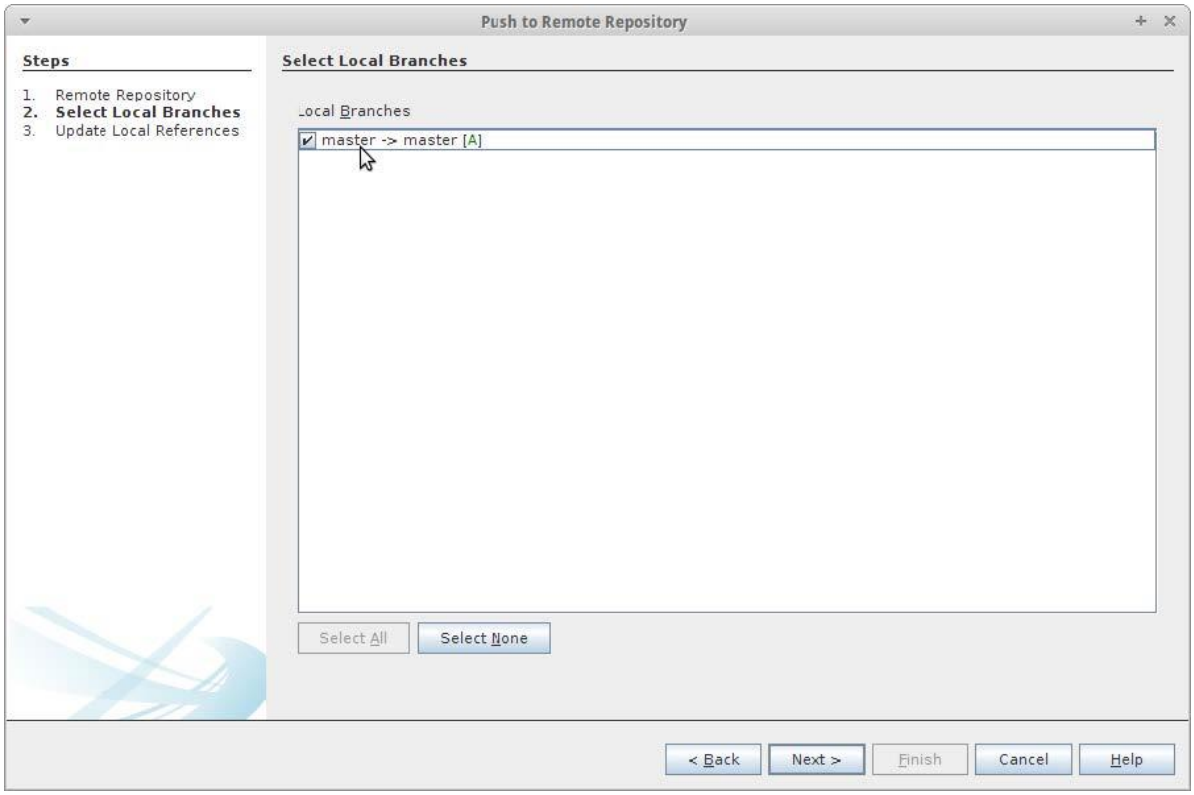


You will then be presented with the push dialog. For the *first* push only, we need to provide some information on our repository:



1. Under "Repository URL" you need to enter the `https` URL address we noted above in Section 1.
2. Enter your bitbucket username and password.
3. Make sure "Remote name" is "origin" above Repository URL.
4. Click "Next"

You will get an error message about branches. Just tick on the "master" branch, and click "Next" and "Finish":



For your *second* and all later pushes, things are easier. When you select "push", you will find the push dialog has the top radio button selected, which contains the url for your project. As the details of your repository are known, just click "Next", "Next" again, and finally "Finish".
This process pushes your code onto the repository. You can check this by going to your webbrowser, and looking at your bitbucket page. Select the "source" menu on the left, and you will be able to navigate through all the files pushed to the repository, and even view your code. You can also see a list of the commits by selecting "Commits" menu one the left.

3. Assessment

We will be using bitbucket within the assessment for coursework. Some marks are available for correct use of version control, so what counts as correct use? Marks will be available for:

1. A meaningful project name (e.g. coursework-1)
2. A commit identifying when each piece of functionality has been completed.
3. A regular pattern of commits over the time you have worked on the assignment.
4. Meaningful commit messages.

Marks in particular will be lost if you upload all your code in a single commit on the last night. However, a "regular pattern of commits" does not mean you cannot do a lot of work in a short time - however we would expect a commit for each piece of added functionality, to demonstrate a careful development process. A screen shot of the commits list will be required with your submission.
As your repositories are private, you will need to invite the tutor to join your project so we can see your files. To do this:

1. Go to the overview page of your repository on bitbucket
2. On the upper right corner, click the "…" next to “Clone”
3. Then select “Share repository”
4. Enter `huicheng`, click "Add"

Use the version-control system as much as you can. You can use public repositories for your exercise solutions in the practical sessions, for example, and discuss them with friends by inviting them to join your project.