Introduction to Git Version Control

What is Version Control?

Version Control (aka "Revision Control" aka "Source Control") lets you track your files over time. Why do you care? So when you mess up you can easily get back to a previous working version.

But that's just the tip of the iceberg. Version control can do lots more for you and your team. Here are two good references where you can learn more about version control systems (VCS) in general, and about the popular Git VCS:

- http://en.wikipedia.org/wiki/Revision_control
- http://progit.org/book/

What Types of Things Can You Do With VCS?

- Keep track of every change made to every file in a project folder and be able to revert back to any previous version
- Create branches to experiment with project changes that may or may not be used in the future – and then merge them with your master branch when and if you're ready
- Merge changes or additions from other team members into your master or other branch
- Clone a VCS repository and work on your own fork
- Store all project files on a remote server and have access to them wherever you go – no need to carry around your USB drive anymore!
- And much, much more

Why Git?

Git is one of the most advanced and most popular VCS in the IT industry today. It was created by Linux inventory Linus Torvalds to fix problems with traditional VCS like Subversion, et al, and to create a system that could be distributed, without the need for a central repository server. Today, Git is arguably the most popular VCS in existence and available free of charge for most computing platforms.

In addition, Git users can take advantage of the popular Github.com website and host their Git repositories for free when using the open-source model.

What Do I Need to Get Started?

Beginning with Netbeans 7.1, support for Git is built into the IDE. That means that for basic Git functions you do not need any other software to start using Git. However, should you need more advanced features not supported by Netbeans, you would need to download the Git client software and install it on your computer.

You can find the client software for your O/S at: http://git-scm.com/

CAUTION: Installing the client software can be tricky, especially if you're running the Windows O/S. Supporting these client installations is beyond the scope and time available in this course. Please be advised that if you install the software, neither your instructor nor anyone else at WCTC will be able to help you troubleshoot problems. **You are advised to stick with the Netbeans support** until you learn more about Git and are able to support yourself.

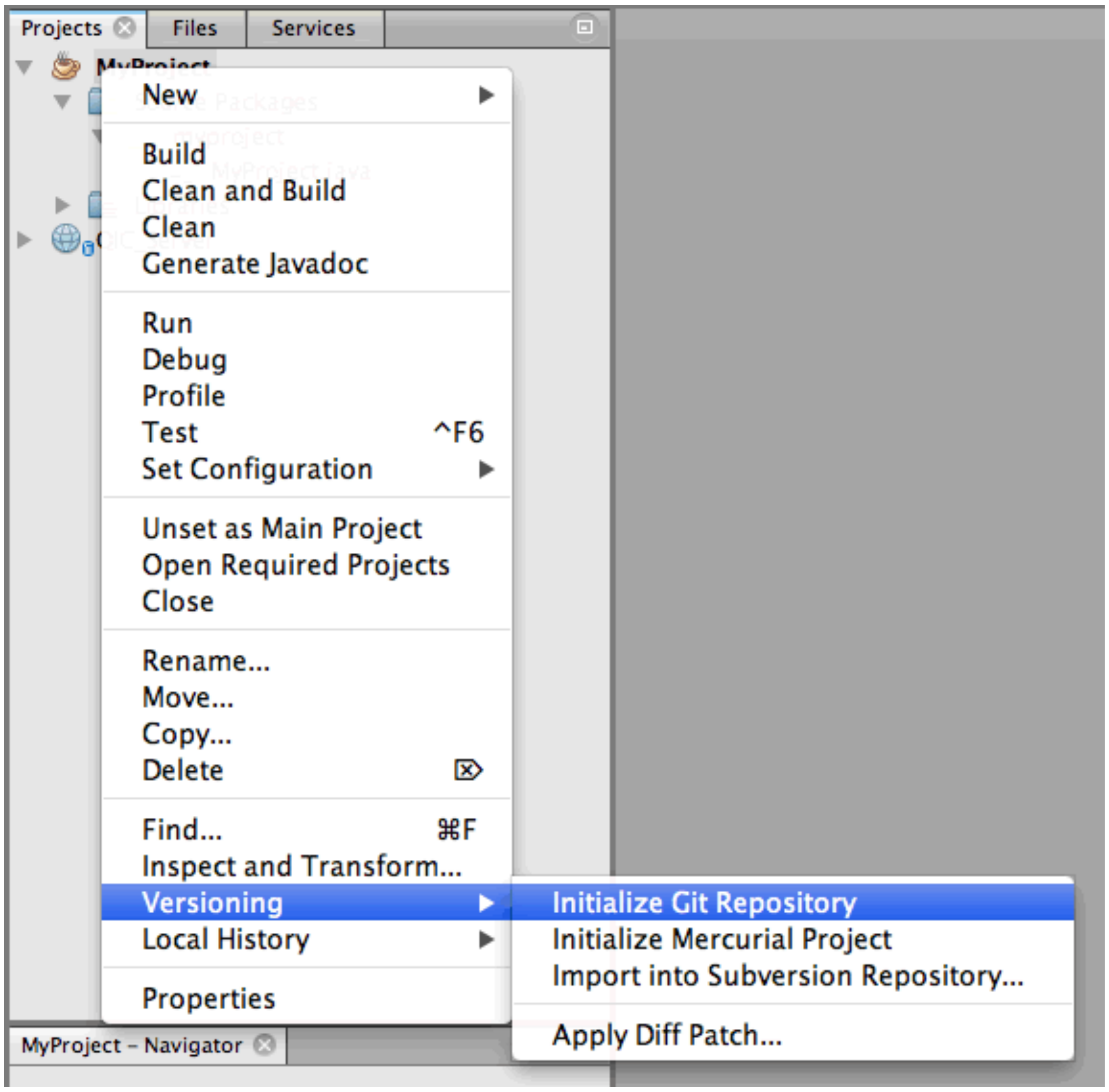
Instructions for Using Git With Netbeans 8.0+

Whether you are starting a new project or want to put an existing project under Git source control, Netbeans makes it easy. We'll provide some simple instructions here, but for more information please refer to this Netbeans tutorial: http://netbeans.org/kb/docs/ide/git.html

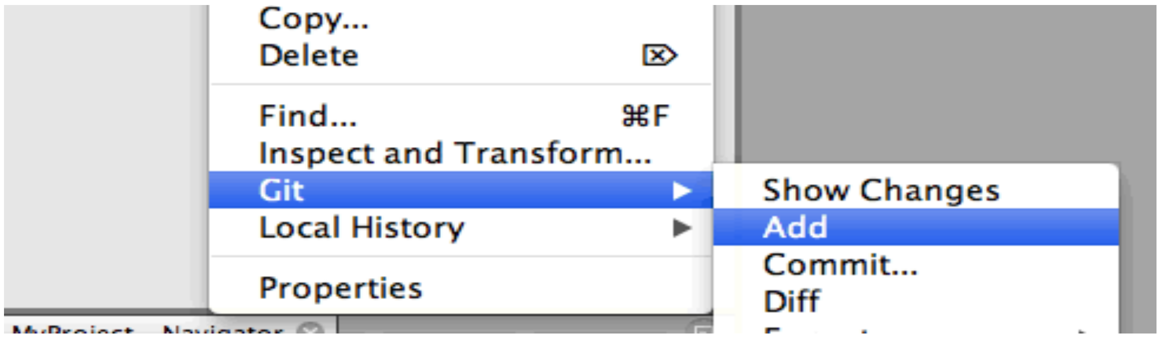
CAUTION: Many Git tutorials, including the one at Netbeans.org, refer to SSH as a secure way of communicating with a remote Git server. This is an older and more complicated technique that should be avoided in favor of using HTTPS. Please ignore any references to SSH-based techniques and instead use the HTTPS techniques provided in this guide.

Putting a new or existing project under Git Source Control:

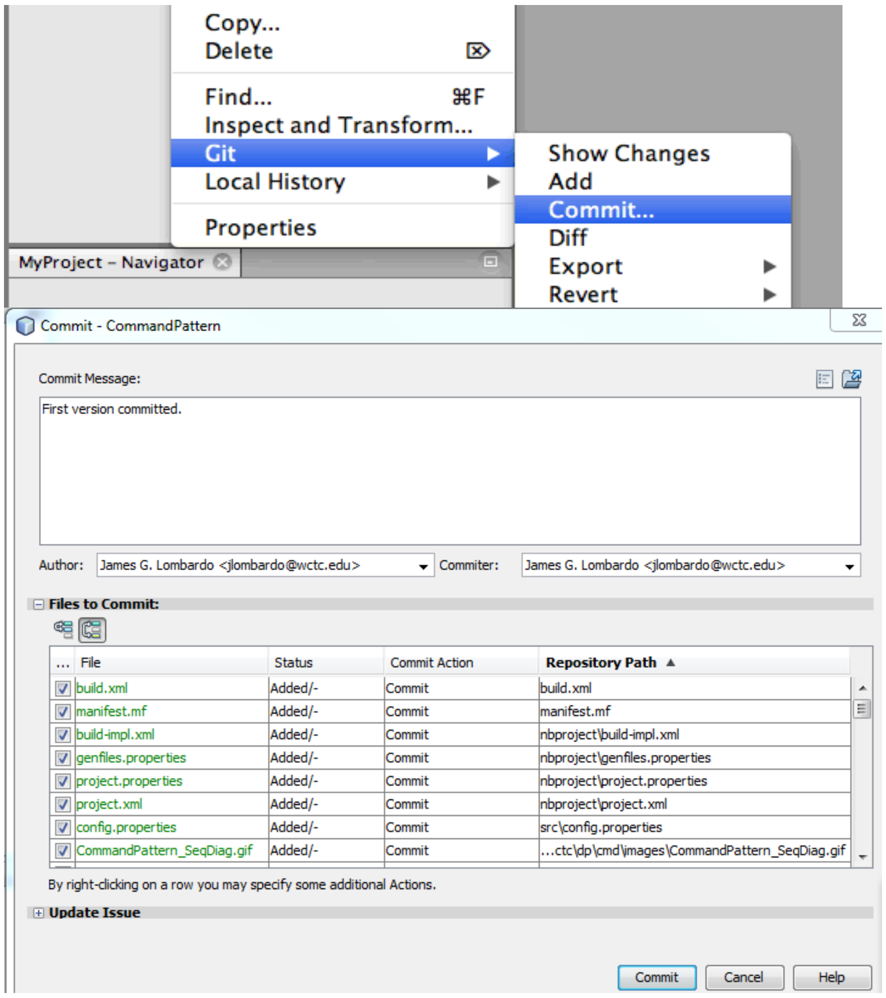
1. Right-click your mouse on the project name in Netbeans and select Versioning > Initialize Git Repository



2. Next, a dialog box pops up showing you where the Git repository will be stored. By default this should be the project folder. Confirm this by clicking the "Ok" button.
3. If this project has just been initialized for Git then right-click on the project in Netbeans and select Git > Add from the popup menu. This will add all of your project files to the Git repository (an invisible database stored within your project folder). However, these files are in what's called a staging area and must be committed to be saved permanently.



4. Next, right-click on the project again and select Git > Commit from the popup menu. You should enter a commit message explaining the additions or changes you made and then commit those changes. This commit message is very important as it will help you in the future if you need to identify a particular commit for any reason, such as reverting back to that commit.



Commit Message:

First version committed.

Author: James G. Lombardo <jlombardo@wctc.edu>
Commiter: James G. Lombardo <jlombardo@wctc.edu>

Files to Commit:

...	File	Status	Commit Action	Repository Path ▲
✓	build.xml	Added/-	Commit	build.xml
✓	manifest.mf	Added/-	Commit	manifest.mf
✓	build-impl.xml	Added/-	Commit	nbproject\build-impl.xml
✓	genfiles.properties	Added/-	Commit	nbproject\genfiles.properties
✓	project.properties	Added/-	Commit	nbproject\project.properties
✓	project.xml	Added/-	Commit	nbproject\project.xml
✓	config.properties	Added/-	Commit	src\config.properties
✓	CommandPattern_SeqDiag.gif	Added/-	Commit	...ctc\dp\cmd\images\CommandPattern_SeqDiag.gif

By right-clicking on a row you may specify some additional Actions.

Update Issue

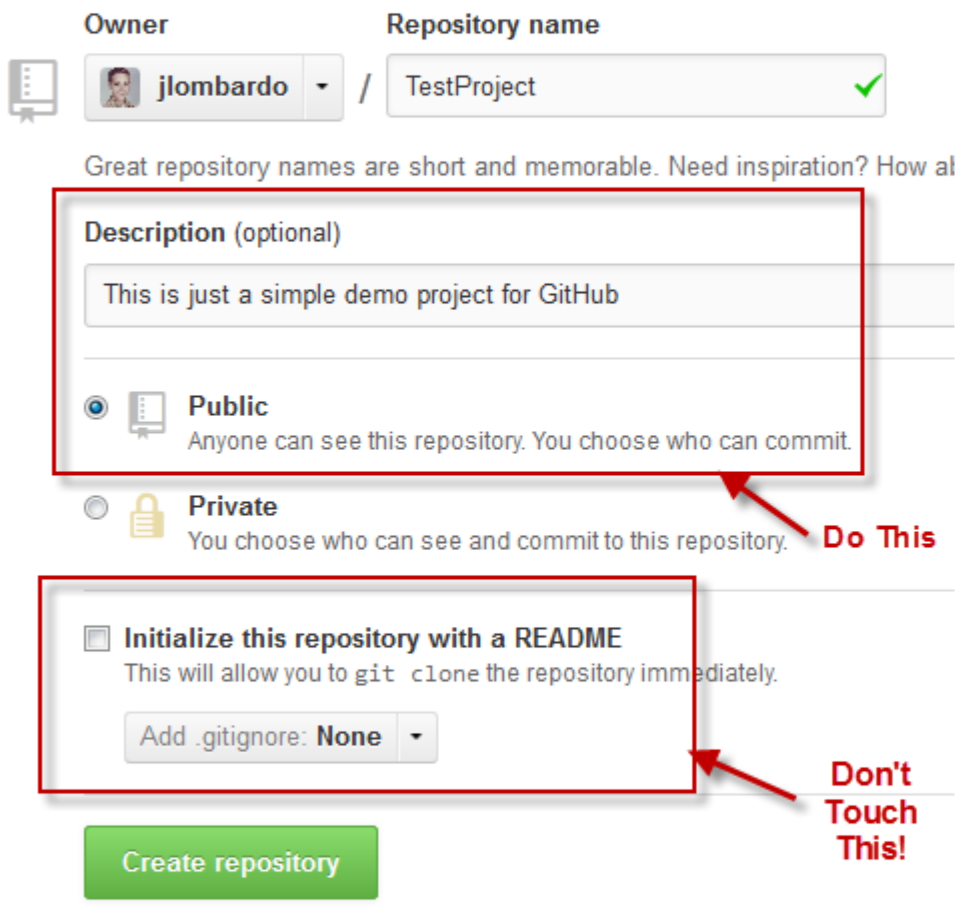
5. Notice the commit message dialog box also shows all the files that will be added to your repository. Further, it shows who the author and commiter are by email address. If your email address is not showing up here, please add it to identify who did the commit.
6. Those are the basics for getting started but there is so much more you need to know about using Git. **It's very important you read this:** how to use Git in Netbeans please visit http://netbeans.org/kb/docs/ide/git.html

How to Store and Maintain Your Git Projects on GitHub

GitHub is a web site that provides free hosting for open-source Git projects. Please be advised that open-source projects are available for anyone to see and download. If you need privacy, you will have to pay a fee to GitHub.

The advantage of hosting your Netbeans project on GitHub is that you do not need to carry around your projects on your USB drive anymore. You can simply work on your project at school and then push all your changes up to the GitHub server. Then when you get home, open the local Netbeans project that you have stored on your computer and pull down the changes. Or, if you created your project at school and do not have a local copy at home, you can clone the project from GitHub.
Step-by-Step Instructions:

1. First you will need to sign up for a GitHub account. From your web browser go to https://github.com/ and click the "Signup and Pricing" link at the top of the page.
2. At the next screen click the "Create a Free Account" button and fill out the signup form. **IMPORTANT: please use your WCTC email address!** Your instructor needs this to track your work.
3. Once you have an account and are logged in, you can create a repository. "Repository" is the term Git uses to managed a project. This includes all of the contents of your Netbeans project folder, or any folder that you want Git to version control.
4. You should see a button labeled "New Repository" that you will click to create it. **When you do this, please use the exact same name as you used for your Netbeans project – with the exact letter case.**
5. Now enter the name of your repository (remember, use the exact same name as your Netbeans project), then enter a description and leave the other settings untouched.


Owner

Repository name

jlombardo / TestProject

Great repository names are short and memorable. Need inspiration? How al

Description (optional)

This is just a simple demo project for GitHub

[✓ Public] Anyone can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with a README

This will allow you to `git clone` the repository immediately.

Add .gitignore: None

6. After your repository is created, GitHub will display a series of command line tasks -- **IGNORE THESE! We will use Netbeans instead.**
7. Leave your web browser connected to GitHub running in the background and switch back to Netbeans. Make sure your Netbeans project is open and has been initialized for Git, and you have added and committed all of your files and/or changes.
8. Now you can "push" those changes to GitHub. To do this, right-click on your project in Netbeans and select
   Git > Remote > Push
9. A dialog box will be displayed. Select the second radio button labeled "Specify Git Repository Location:" and enter the "Repository URL:" in this format, where "user_name" is your GitHub user name and "Project_Name.git" is your project name with a ".git" extension:

   ```
   https://github.com/user_name/Project_Name.git
   ```

   ... then fill in your GitHub username and password and click "Next"

10. **IMPORTANT: If at any time AFTER the previous step you see a small dialog box popup with ONLY username and password – make sure you**

press CANCEL. This dialog box should not be used and will cause your PUSH operation to fail. This is a bug in Netbeans.

11. Follow all remaining prompts to complete the process. If you see a window that lists branches to publish, click all of them. Your code is now on GitHub! Go back to your GitHub project in your web browser and select it or refresh the screen. Notice that all of your files are now on GitHub.
12. In the future when you make changes to your Netbeans project all you will need to do is add, commit and then push to update your GitHub respository.
13. **NEVER, EVER make changes to your project from within GitHub** on your web browser. Treat the GitHub content as the master that you only update from Netbeans.