

Eclipse EGit GitHub

How to setup Eclipse Git Plugin EGit for GitHub Part 1 & 2 [Tutorial]



<http://web.archive.org/web/20140625082056/http://benjsicam.me/blog/how-to-setup-eclipse-git-plugin-egit-for-github-part-1-tutorial/>

<http://web.archive.org/web/20140625122619/http://benjsicam.me/blog/how-to-setup-eclipse-git-plugin-egit-for-github-part-2-tutorial/>

Voir aussi http://wiki.eclipse.org/EGit/User_Guide

How to setup Eclipse Git Plugin EGit for GitHub Part 1 [Tutorial]



Posted on May 26, 2013 in [Blog](#), [Development Environment](#), [Eclipse](#), [Git](#), [IDE](#) | [0 comments](#)

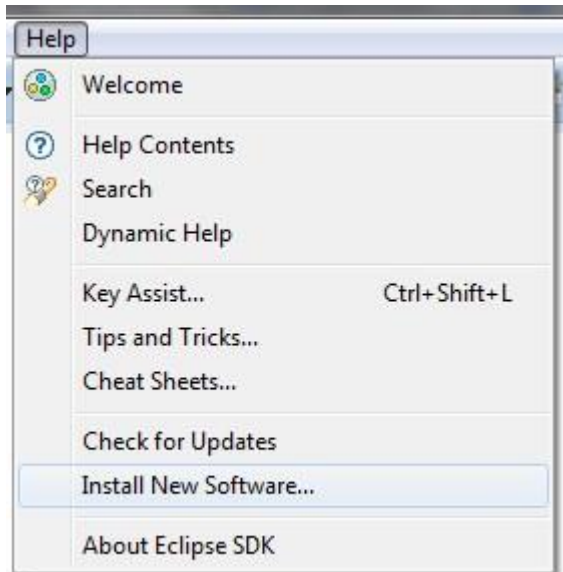
This tutorial is all about Eclipse and the EGit plugin which is used to connect Eclipse to Git – a distributed version control system (DVCS). This tutorial will specifically discuss on how to connect your Eclipse to GitHub with EGit.

What we need:

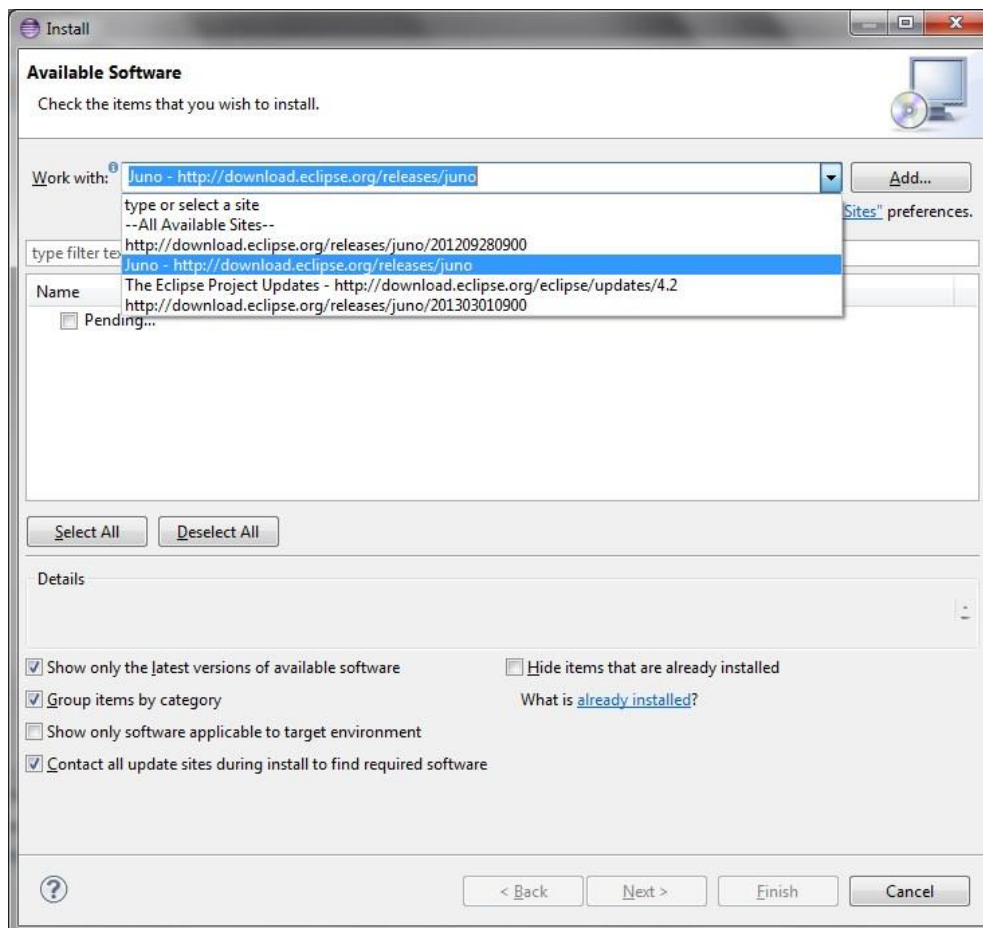
1. [Eclipse Classic](#). Current version 4.2.2 (Juno).
2. EGit plugin. Current version: 2.3.1.201302201838-r
3. A GitHub Account. If you do not have one, please [sign up](#) on GitHub. This tutorial assumes that you already have a GitHub account.

Part 1 – Installing EGit on Eclipse

1. Download Eclipse Classic from <http://www.eclipse.org/downloads/>
2. Unpack the zip file anywhere on your Windows PC.
3. Run Eclipse by double-clicking eclipse.exe.
4. Set your workspace path when asked.
5. When the Eclipse Dashboard opens, navigate to Help > Install New Software



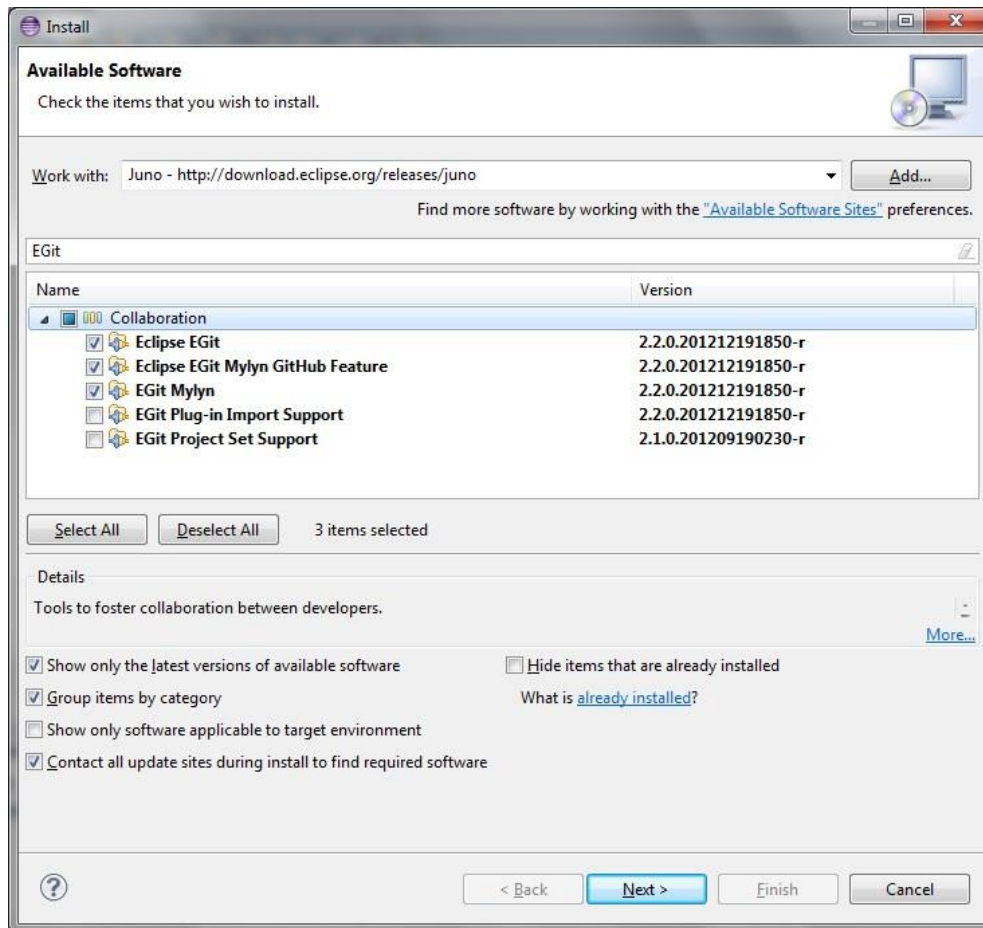
6. On the Install window, click the Work With drop-down and select “Juno – <http://download.eclipse.org/releases/juno>”



7. After selecting the site, it will look for the software/plugins available for download. Wait for a moment until Eclipse has successfully searched all of them.

8. Type EGit on the filter text and hit the Enter key.

9. Check Eclipse EGit, Eclipse EGit Mylyn GitHub Feature, and EGit Mylyn.



10. After that, click Next and follow the setup wizard.

With the EGit plugin already installed on your Eclipse, you will now be able to connect your Eclipse to your GitHub account. The [next part](#) of this tutorial will guide you through setting up the integration

Now that you've already setup EGit on your Eclipse, it is time for you to connect to your GitHub account using it.

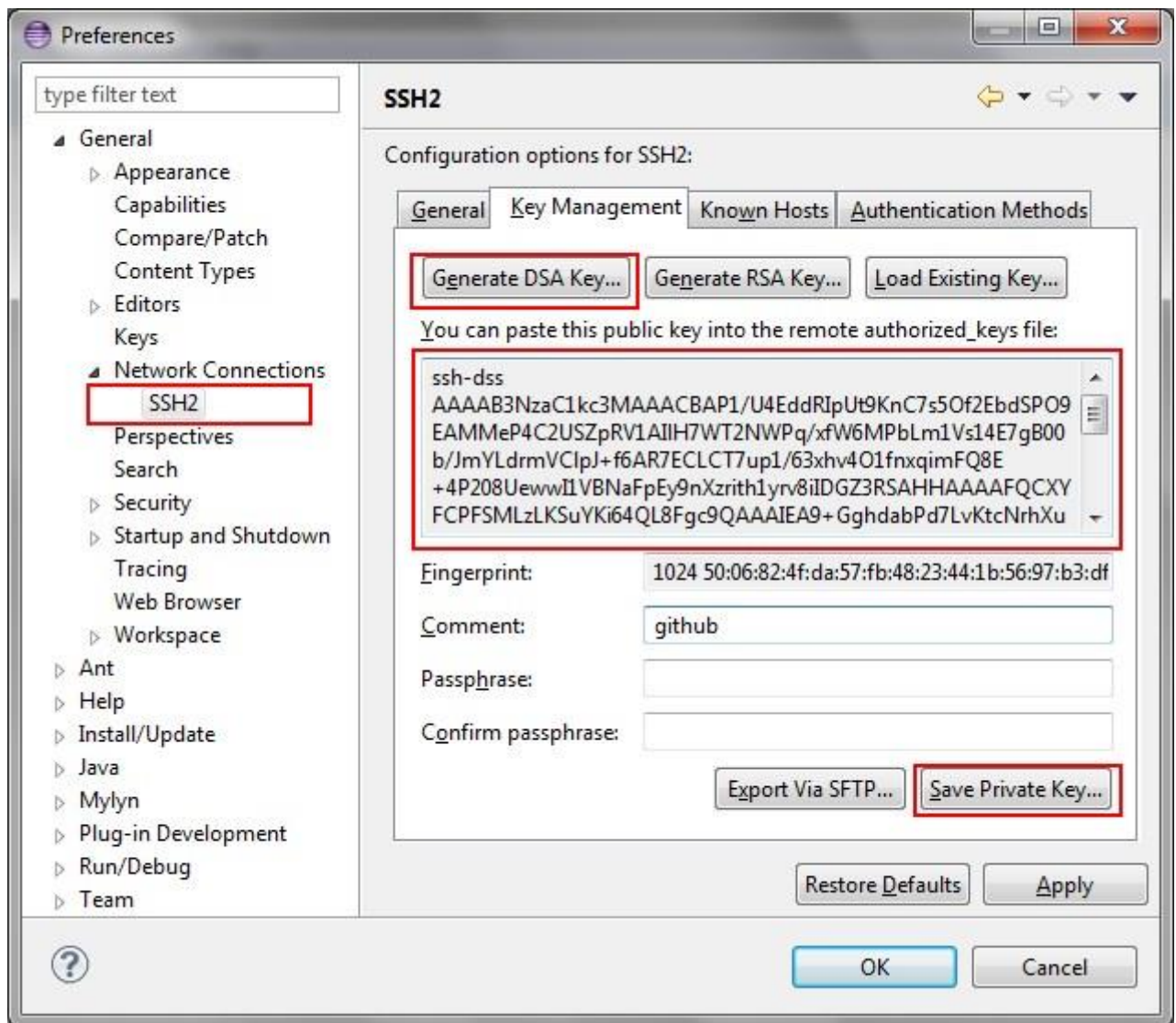
Part 2 – Setting up Eclipse and GitHub Integration

Setup HOME environment variable.

1. Open Start > Right Click Computer and click Properties.
2. Click Advanced System Settings.
3. Open Environment variables.
4. Add a user variable HOME and set it to your user folder e.g. C:\Users\yourusername

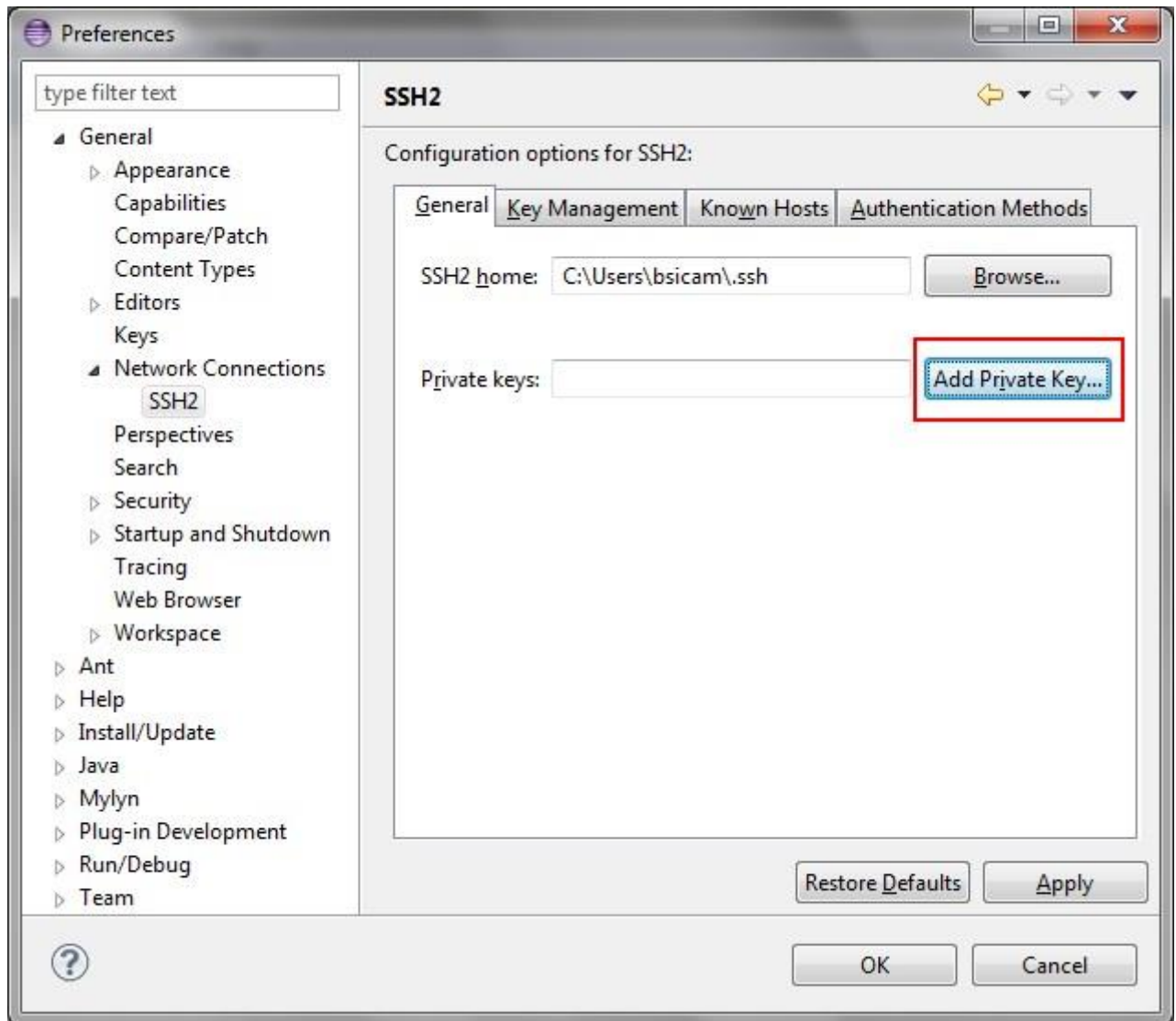
Create SSH keys (private and public keys).

1. Open Eclipse and navigate to Window > Preferences.
2. Go to General > Network Connections > SSH2.
3. Go to Key Management tab and then click Generate DSA key.
4. Copy the generated public key and paste it onto your preferred text editor.
5. Click the Save Private Key button and save the private key under C:\Users\yourusername\.ssh folder (if you are a windows user).



Setup the private key on Eclipse.

1. While still being on the Eclipse Preferences (General > Network Connections > SSH2), go to the General tab.
2. Click the Add Private Key button.
3. Browse for the private key you've saved under the C:\Users\yourusername\.ssh folder.



Setup the public key on your GitHub account.

1. Sign in to your GitHub account.
2. Go to Account Settings > SSH Keys
3. Click Add SSH Key.
4. Type any Title and copy the public key you've pasted on the text editor and paste it on the Key field and click Add Key.

benjsicam

- Profile
- Account Settings
- Emails
- Notification Center
- Billing
- Payment History
- SSH Keys**
- Security History
- Applications
- Repositories
- Organizations
- PRIVATE REPOS 0 OF 0

Need help? Check out our guide to [generating SSH keys](#) or [troubleshoot common SSH Problems](#)

SSH Keys [Add SSH key](#)

Dev (2b:f8:69:aa:cd:c7:b1:1f:fa:69:63:9a:77:cd:83:73)	Delete
---	------------------------

Add an SSH Key

Title

Key

[Add key](#)

Setup a new repository on GitHub.

1. Click the create new repo link.
2. Provide a repository name and a description and click Create Repository.
3. Now you will be given an SSH URI to your repository which looks like this: `git@github.com:benjsicam/your_repository_name.git`
4. Copy that and head to the next step.

Owner: benjsicam / Repository name: ✓

Great repository names are short and memorable. Need inspiration? How about **drunken-octo-tribble**.

Description (optional):

☒ **Public**
Anyone can see this repository. You choose who can commit.

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

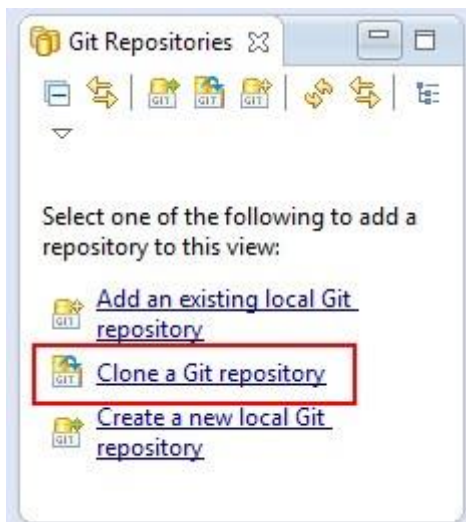
☒ **Initialize this repository with a README**
This will allow you to `git clone` the repository immediately.

Add .gitignore: **None**

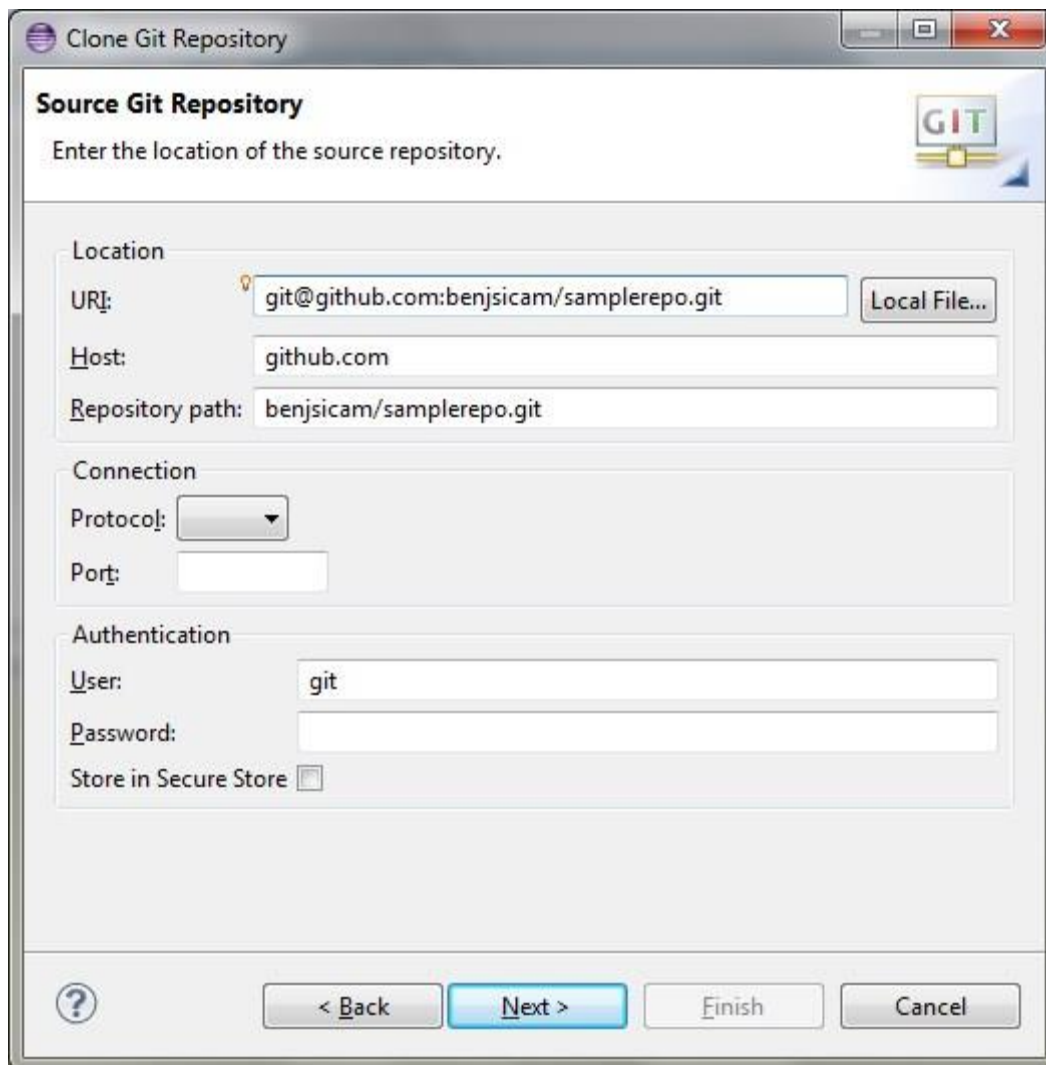
[Create repository](#)

Use EGit to clone the repository and make it a project within Eclipse.

1. Go back to Eclipse and navigate to Window > Show View > Other
2. Under the Git folder, choose Git Repositories and click Ok.
3. On the Git Repositories view, click Clone Git Repository.



4. Select URI as the repository source.
5. The URI should be automatically populated with the SSH URI of your git repository. If not, paste your SSH URI on the URI field and hit Next.
6. Follow the wizard instructions along the way. Just check Import all Projects when clone finishes.
7. Finish the wizard.



After following the steps above you can now place code on your repository and commit your code both on the local repository and the remote repository which is hosted on your GitHub account.

If you would like to know more about Git and GitHub, I recommend reading the O'Reilly Media's [Version Control with Git, 2nd Edition](#) book.

If you would like to know about EGit and how you can maximize its use, you may refer to this [Git version control with Eclipse \(EGit\) tutorial by Lars Vogel](#).