Git with Eclipse (EGit) - Tutorial

Lars Vogel

Version 2.7

09.11.2011

Revision History		
Revision 0.1 Created	11.12.2009	Lars Vogel
Revision 0.2 - 2.7 bug fixes and enhancements	12.12.2009 - 09.11.2011	Lars Vogel

Git with Eclipse (EGit)

This tutorial describes the usage of EGit; an Eclipse plugin to use the distributed version control system Git. This tutorial is based on Eclipse 3.7 (Indigo).

Table of Contents

1. Overview

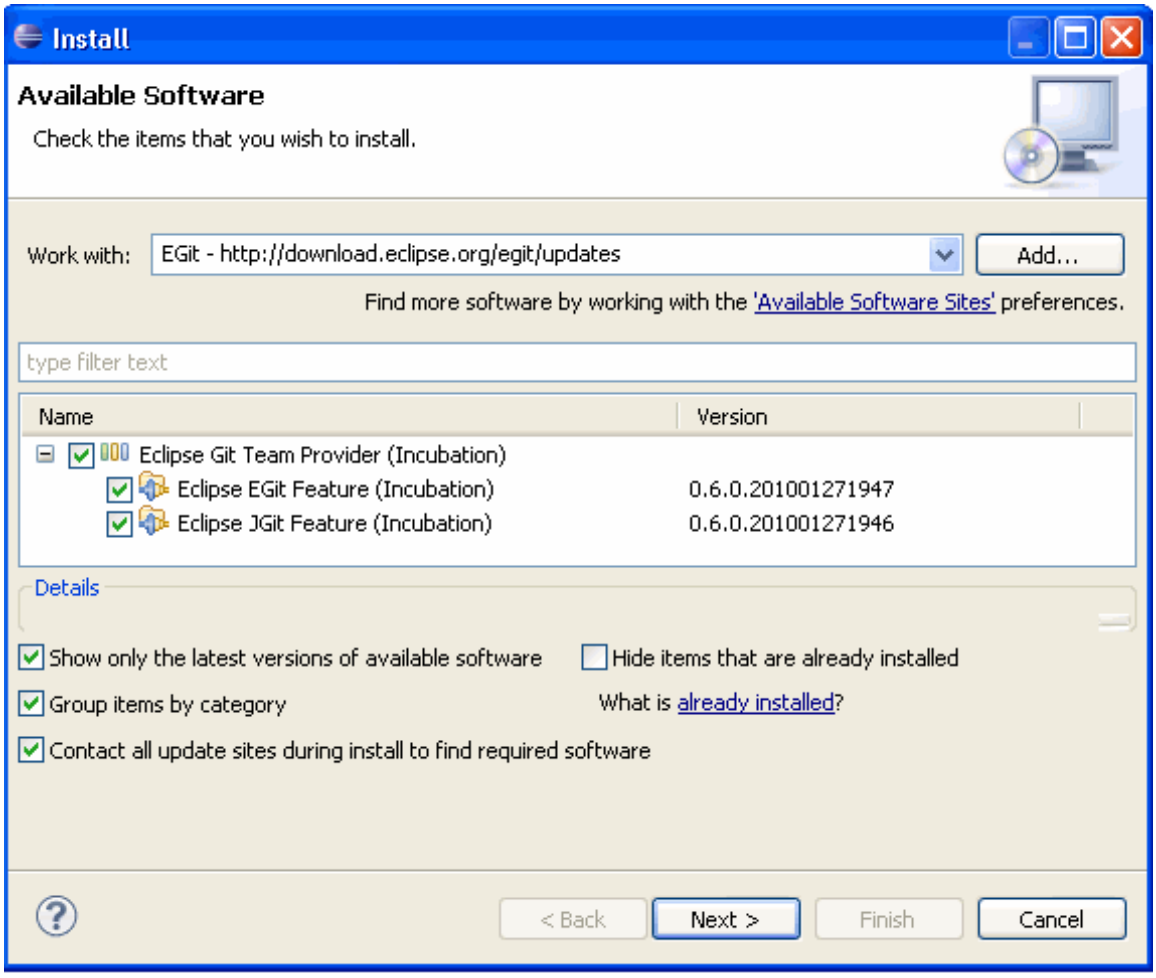
Git is a distributed version control system written in C. JGit is a library which implements the Git functionality in Java. EGit is an Eclipse Team provider for Git (using JGit) and makes Git available in the Eclipse IDE.

This tutorial describes the usage of EGit. If you want to learn about the usage of the Git command line you can use the Git Tutorial as a reference.

2. EGit Installation

Eclipse 3.7 contains EGit in its default configuration. So typically you do not need to install anything.

If you use an older version of Eclipse you can use the Eclipse Update manager to install the EGit plugin from http://download.eclipse.org/egit/updates. The latest features can be found in the nightly build but this build is not as stable as the official update site target http://download.eclipse.org/egit/updates-nightly .



3. Prerequisites for this tutorial

This article assumes what you have basic understanding of development for the Eclipse platform. Please see Eclipse RCP Tutorial or Eclipse Plugin Tutorial .

4. Putting projects under version control

The following section explains how to create a repository for one project and shows how to checkout an exiting projects from a remote repository.

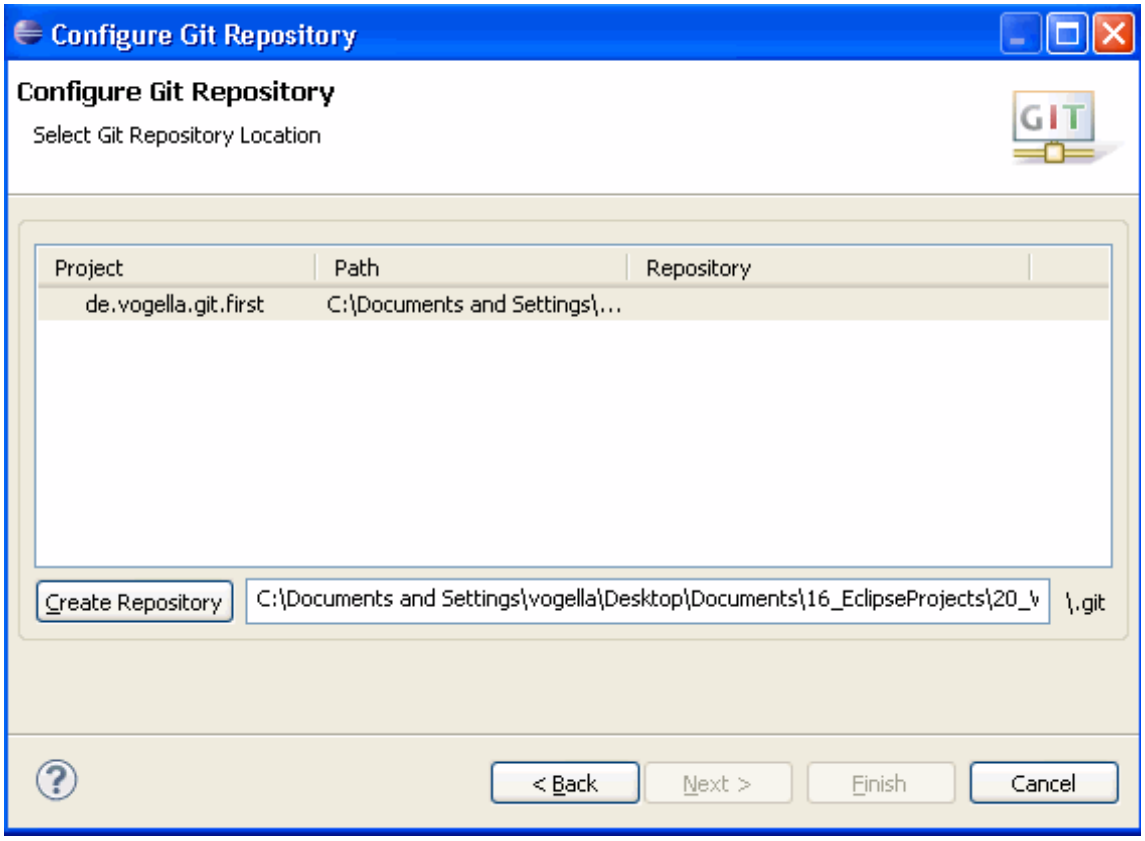
4.1. Put a new project under version control

Create a new Java project "de.vogella.git.first" in Eclipse. Create the following class.

```
package de.vogella.git.first;

public class GitTest {
        public static void main(String[] args) {
                System.out.println("Git is fun");
        }
}
```

Right click your project, select Team -> Share Project -> Git. Select the proposed line and press "Create repository". Press finish.



You have created a local Git repository. The git repository is in this case directly stored in the project in a ".git" folder.

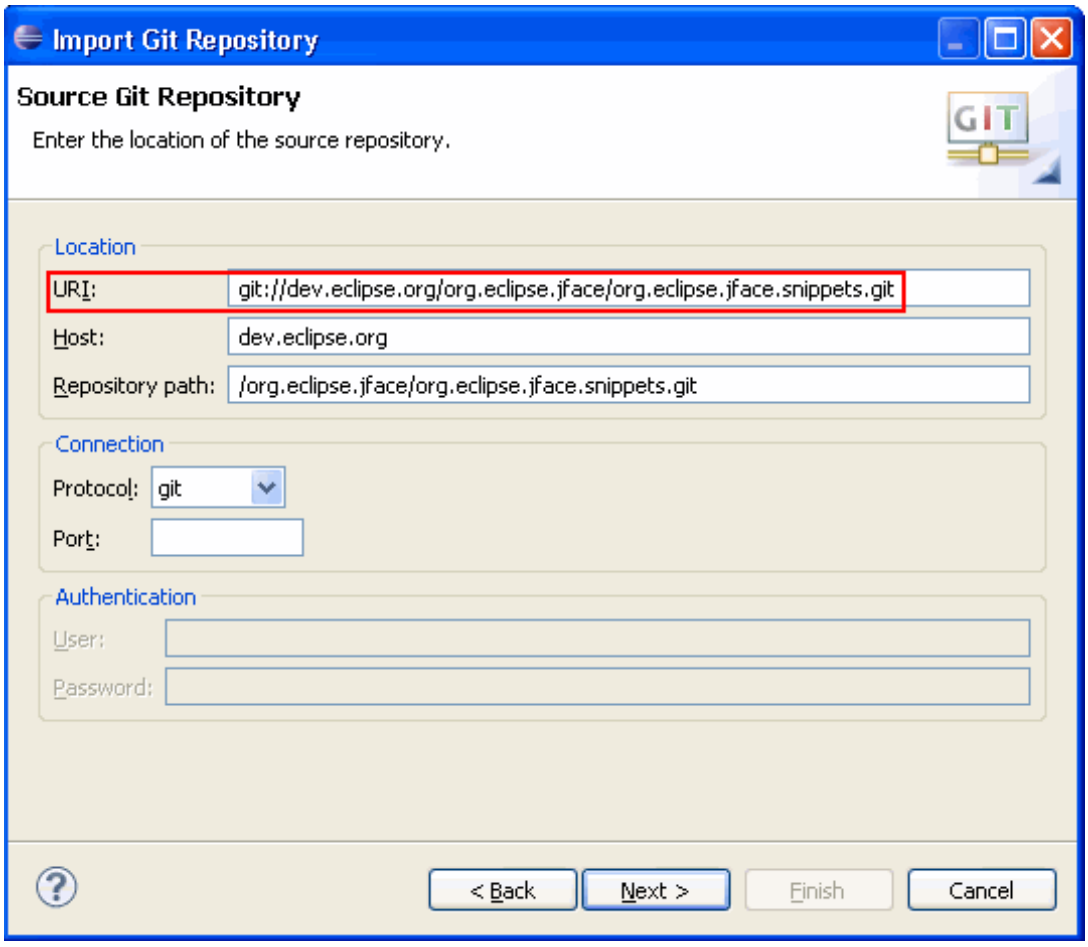
Create the file ".gitignore" in your project with the following content. All files / directories which apply to the pattern described in this file will be ignored by git. In this example all files in the "bin" directory will be ignored.

```
bin
```

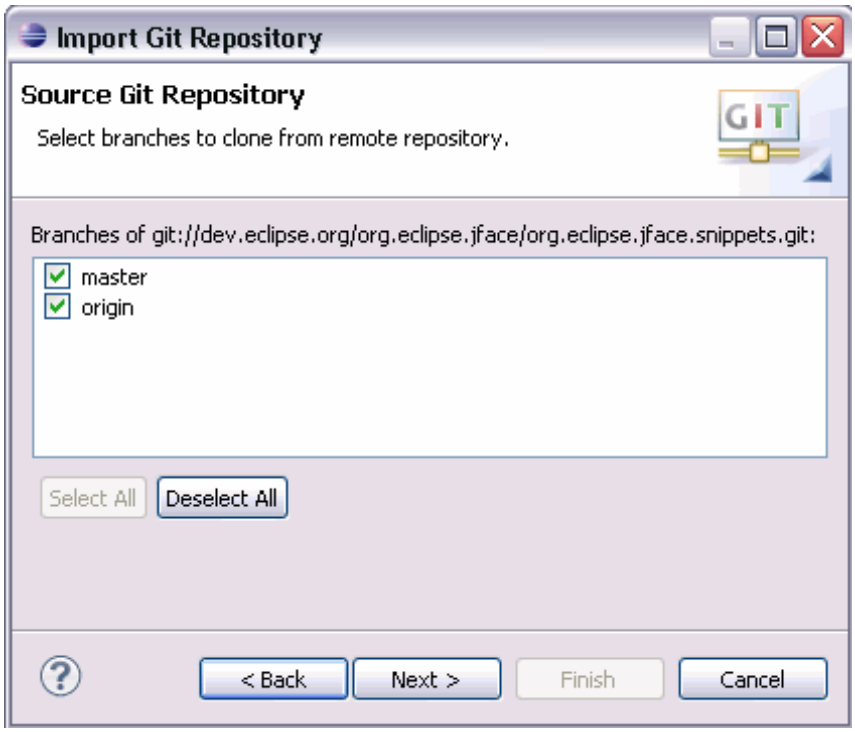
4.2. Clone existing project

EGit allows to clone an existing project. We will use an Eclipse plugin as example. To learn more about accessing standard Eclipse coding please see Eclipse Source Code Guide.

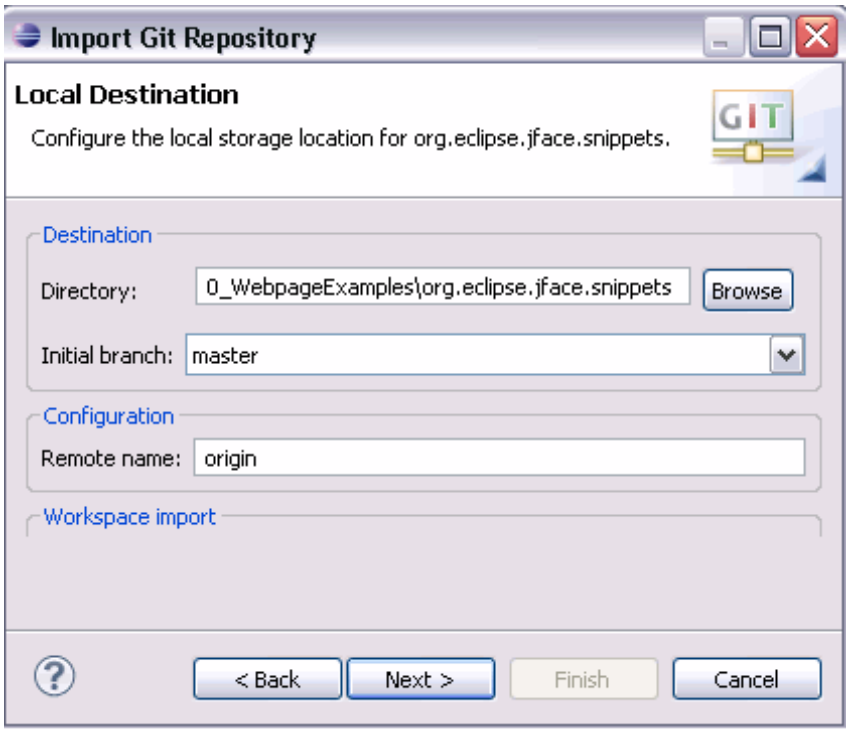
Select File -> Import -> Git -> Git Repository. Press clone and maintain the git repository "git://dev.eclipse.org/org.eclipse.jface/org.eclipse.jface.snippets.git". You only have to paste the URL to the first line of the dialog, the rest will be filled out automatically.



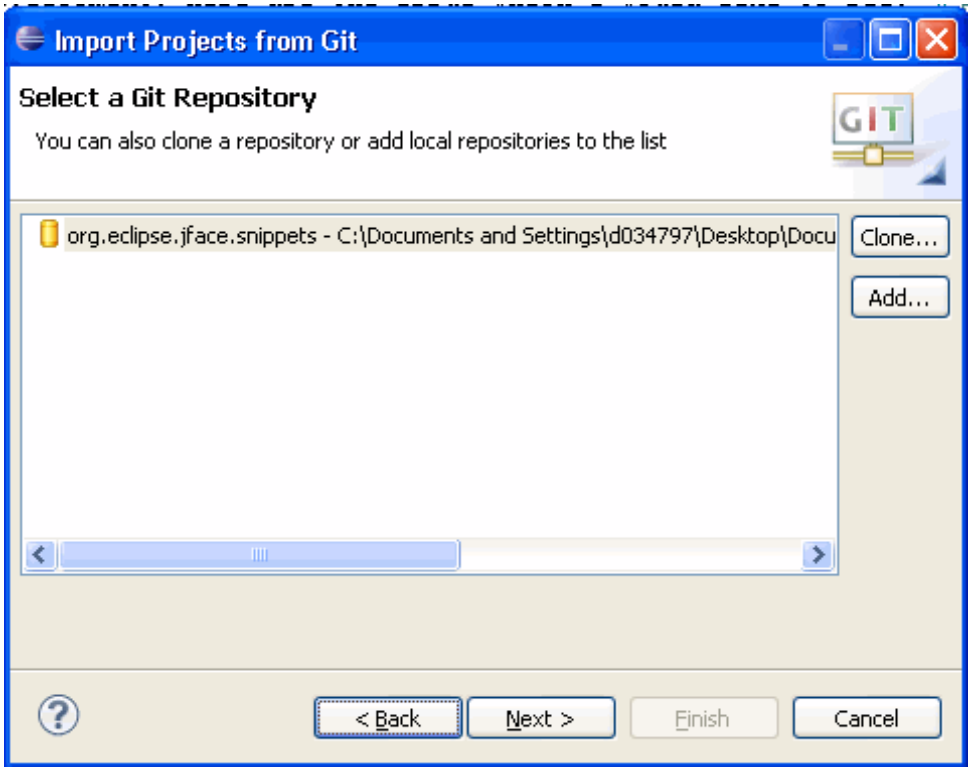
After pressing next the system will allow you to import the existing branches. You should select at least "master".



The next dialog allow you to specify where the project should be copied to and which branch should be initially selected.



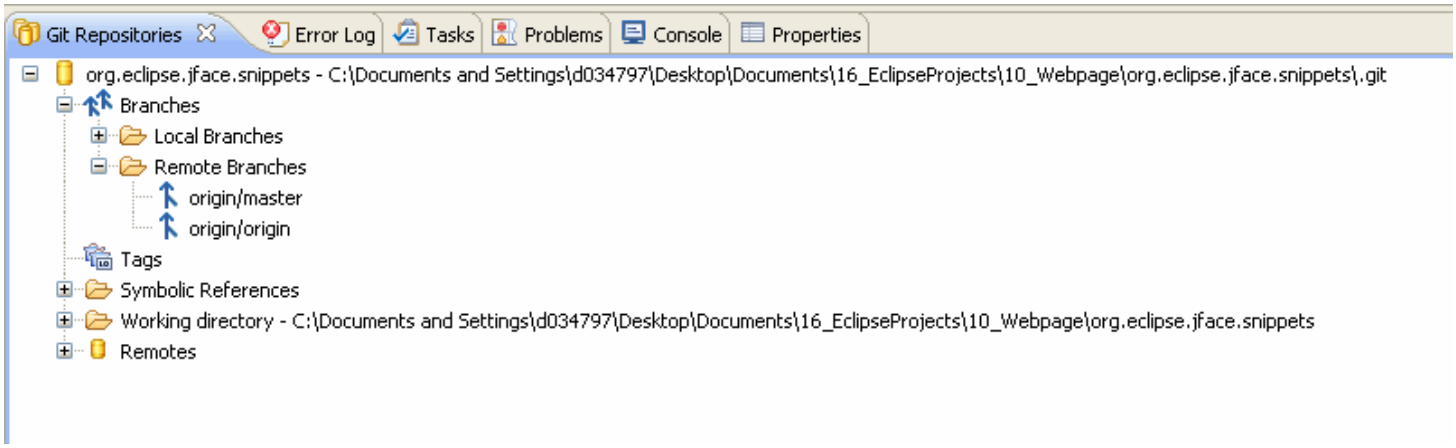
You get an additional import dialog.



You have checked to the project and can use the team Git operation on your project.

4.3. Repository view

EGit has a "Git repository"" view which allow you to browse your repositories, checkout projects and manage your branches.



5. Using EGit

5.1. Basic operations

Once you have placed a project under version control you can start using team operations on your project. The team operations are available via right mouse click on your project. You can:

- Select "Team" -> "Add", on the project node to add all files to version control.

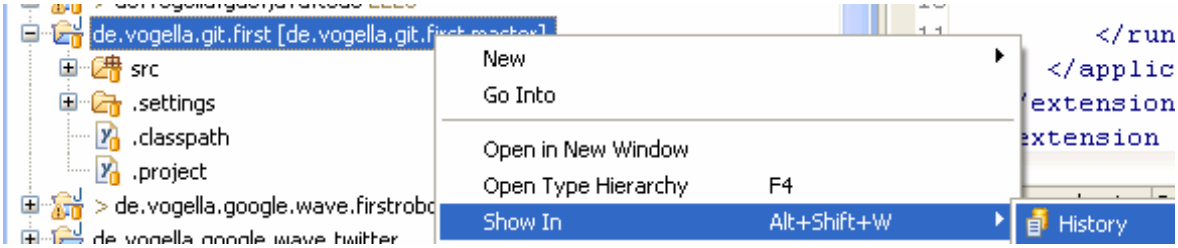
- Select "Team" -> "Commit", to commit your changes to the local Git repository.
- Select "Team" -> "Branch" to create and to switch between branches.
- Select "Team" -> "Push" to push your change to a remote Git repository (see the GitHub chapter).
- "Team" -> "Tag" allows you to create and manage tags.

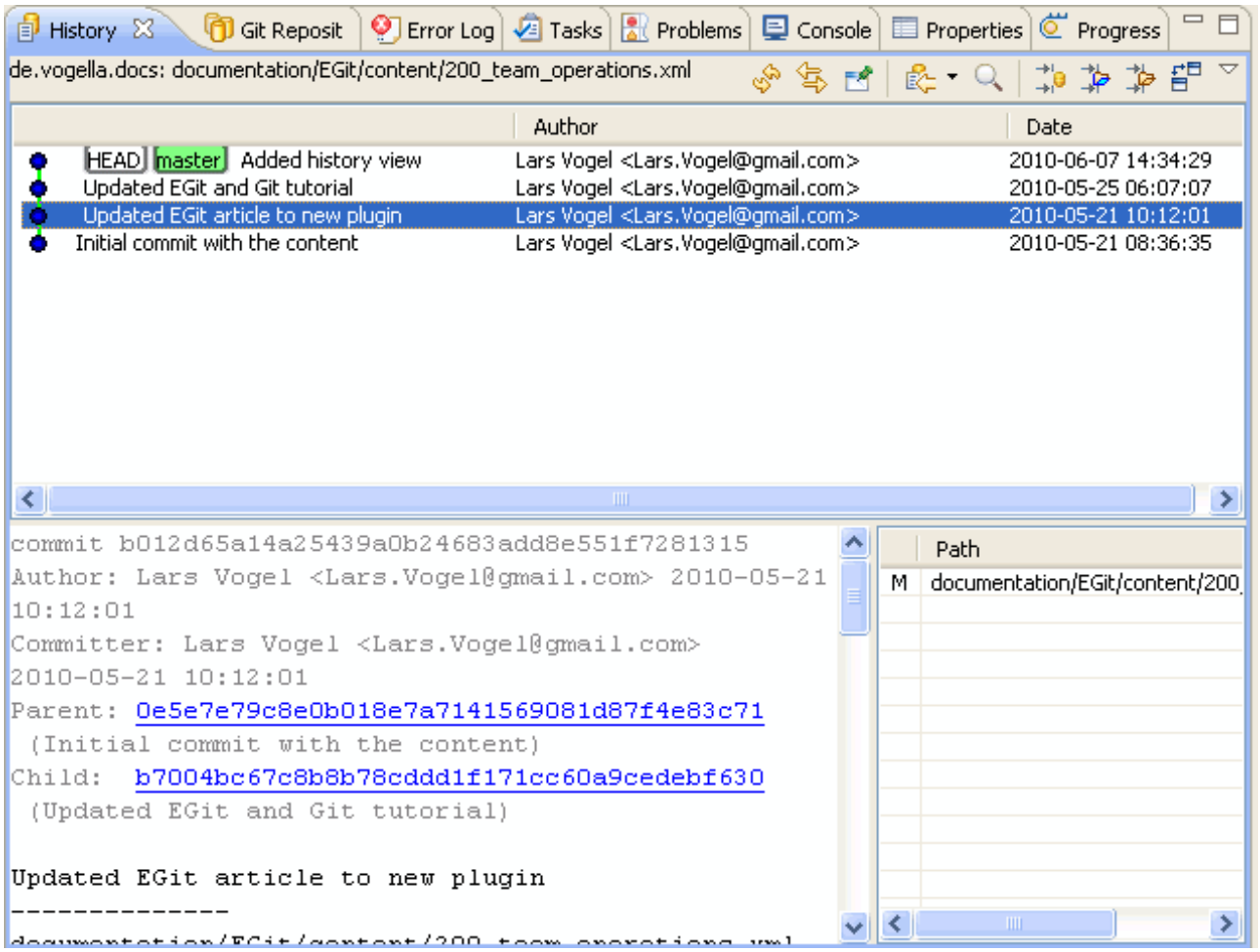
5.2. Merge

EGit supports currently fast-forward merge. Fast-forward merge allows to add the changes of one branch into another if this branch has not been changed. Select "Team" -> "Merge" to perform this operation.

5.3. View the resource history

Select a resource and select Show in (Alt+Shift+W) -> History to see the commit history of this resource.

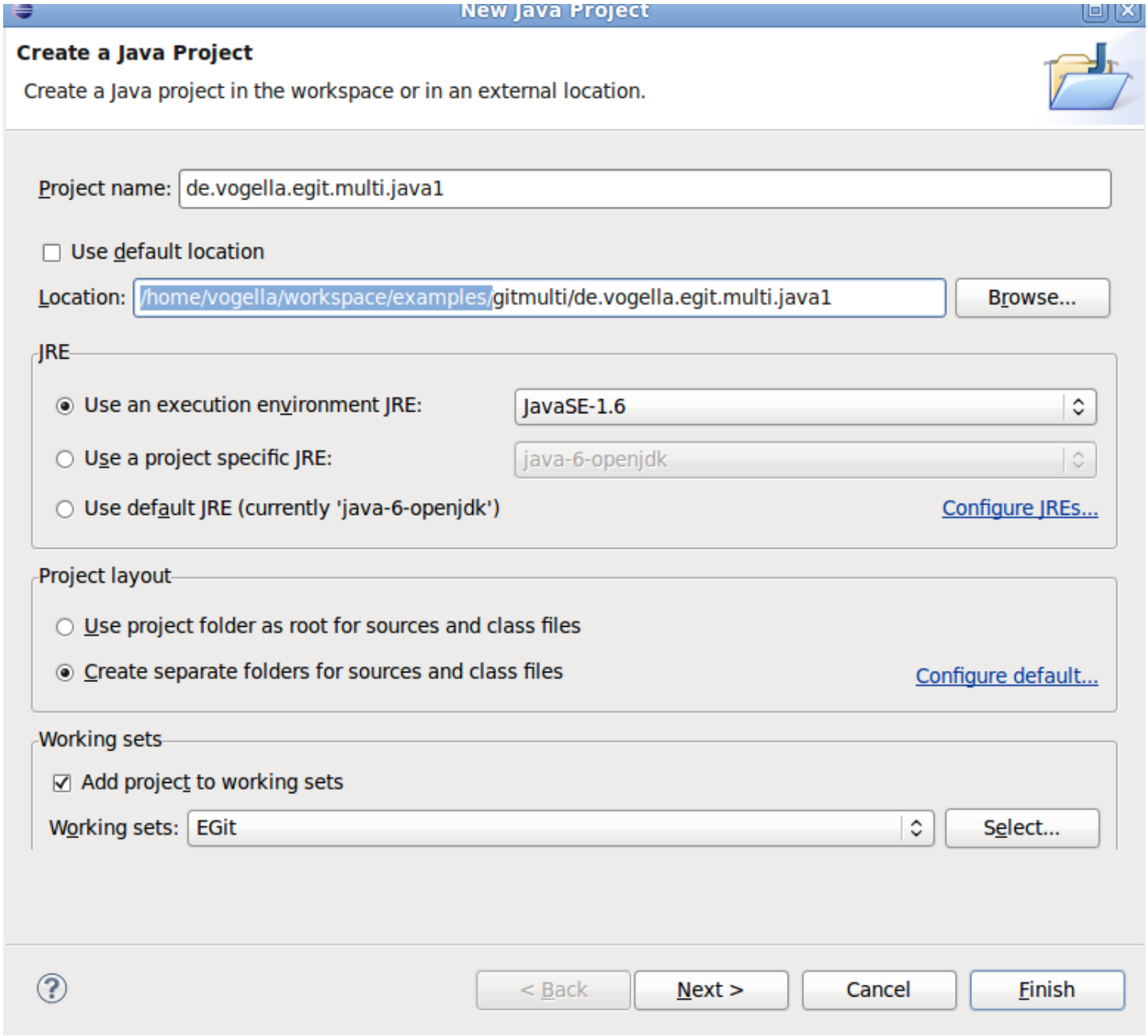


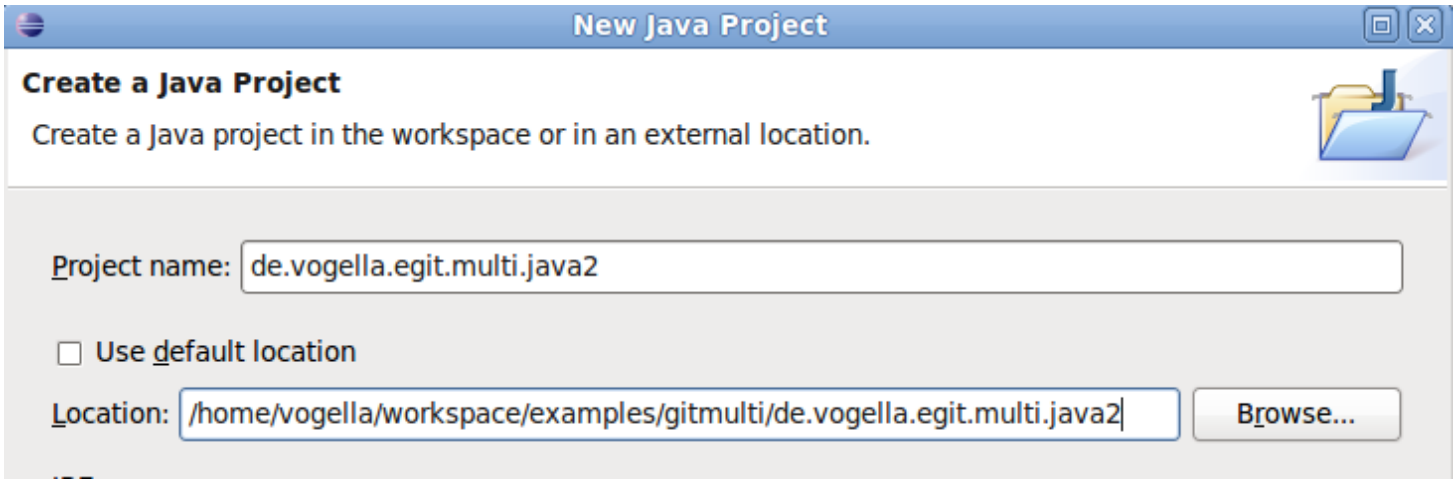


6. Create a Git repository for multiple projects

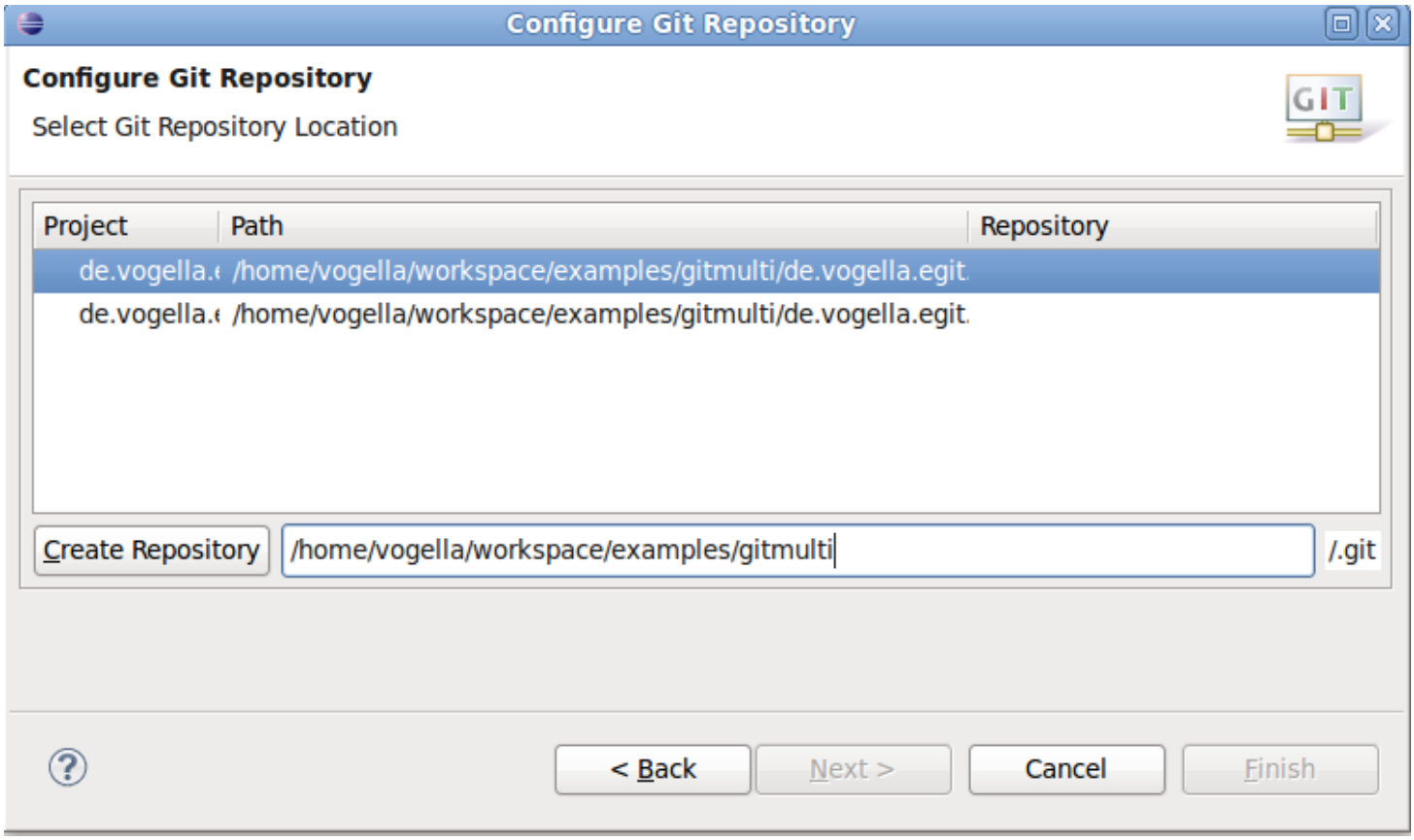
To put several plugins into the same git repository you should create them in a subfolder under your workspace and then create via EGit a repository for this subfolder.

To test this create two new Java projects "de.vogella.egit.multi.java1" and "de.vogella.egit.multi.java2". Do not use the default location (which would be the workspace) but in a subfolder "gitmulti".

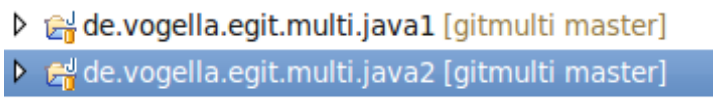




Create a few classes, as git is not able to save empty folders. Now select both projects, right click on them, select Team-> Share-> Git.



Done. Both projects are now in the same git repository.



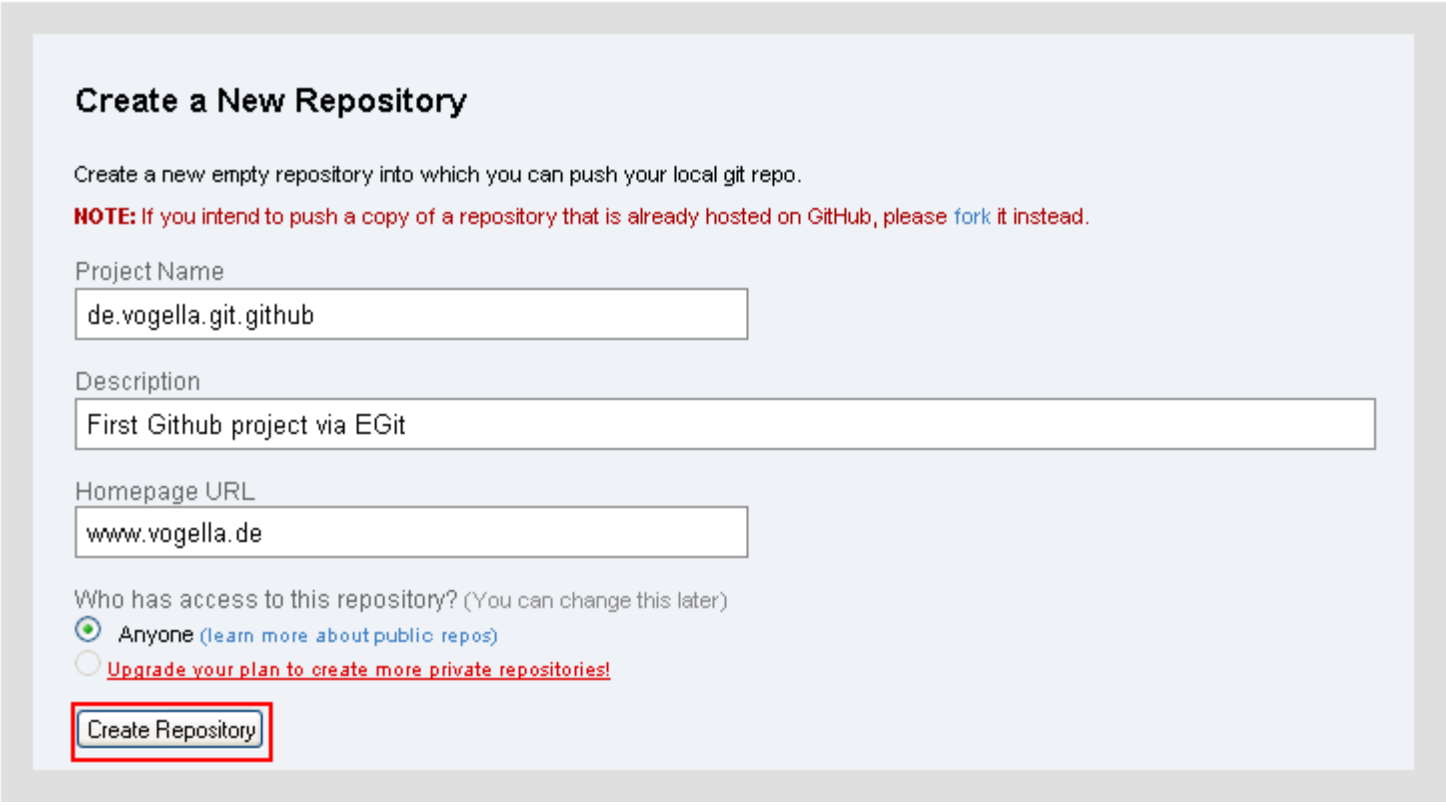
7. Using EGit with Github

7.1. Github

Github is a popular hosting provider for Git projects and if you repository is public to everyone Github hosting is free. To use GitHub create an account on the Github Website. During the sign-up you will be asked to provide a "passphase". This "passphase" is later needed to push to Github from your local repository.

7.2. Create Repository in Github

Create a new repository on Github, e.g. "de.vogella.git.github".

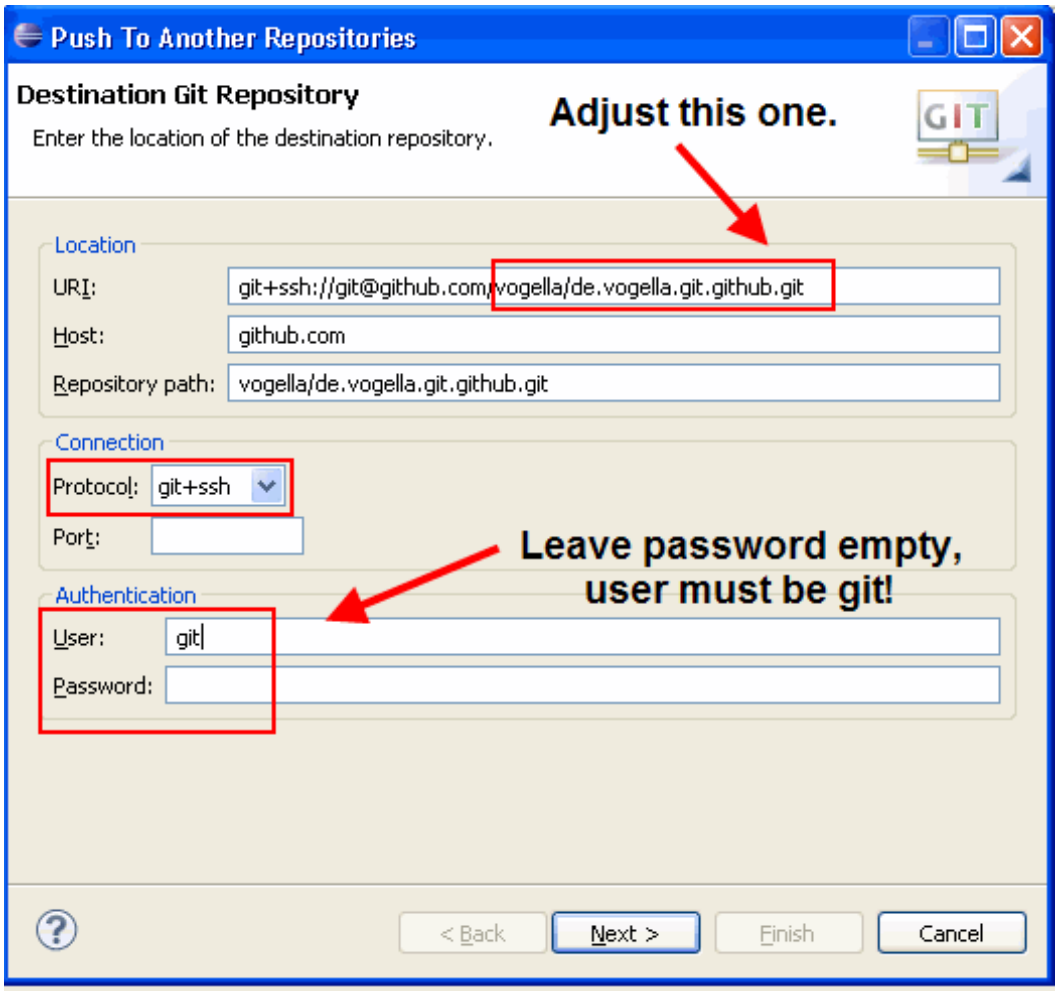


After creation of your new repository Github tells you what to do if you would inport via the command line. As we are going to use EGit we can ignore this information.

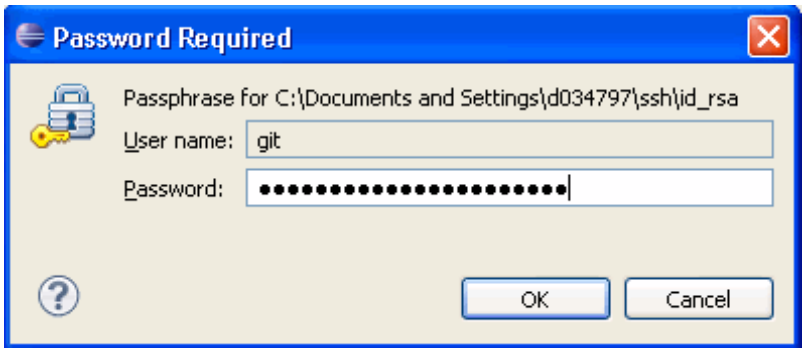
7.3. Push project to Github

Create a new project "de.vogella.git.github" and put it under version control. Right-mouse click on your project and select "Team" -> "Push". A dialog pops up. Maintain the following data. Adjust the hightlighted line so that you are using your user and your project name.

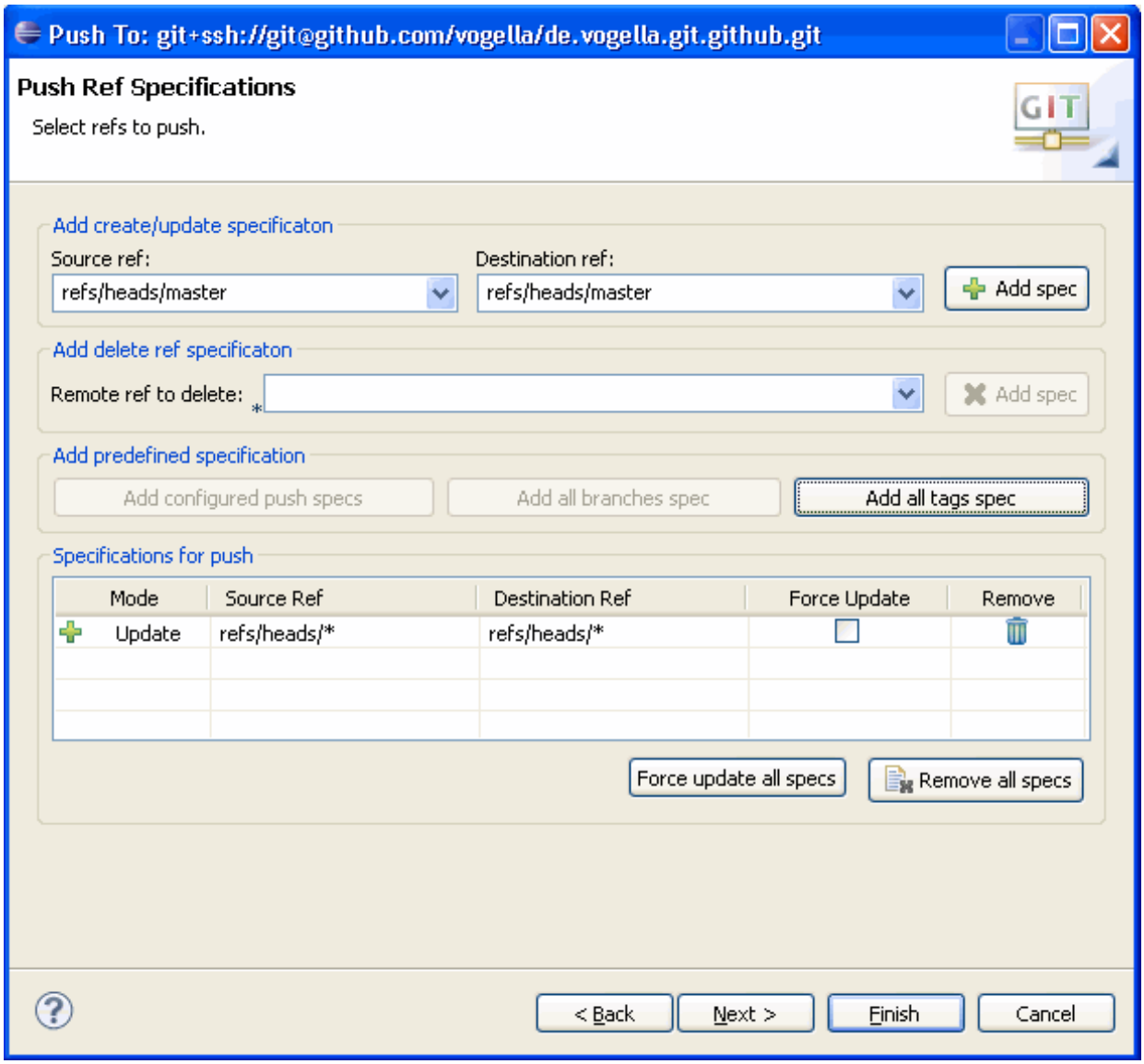
```
git+ssh://git@github.com/vogella/de.vogella.git.github
```



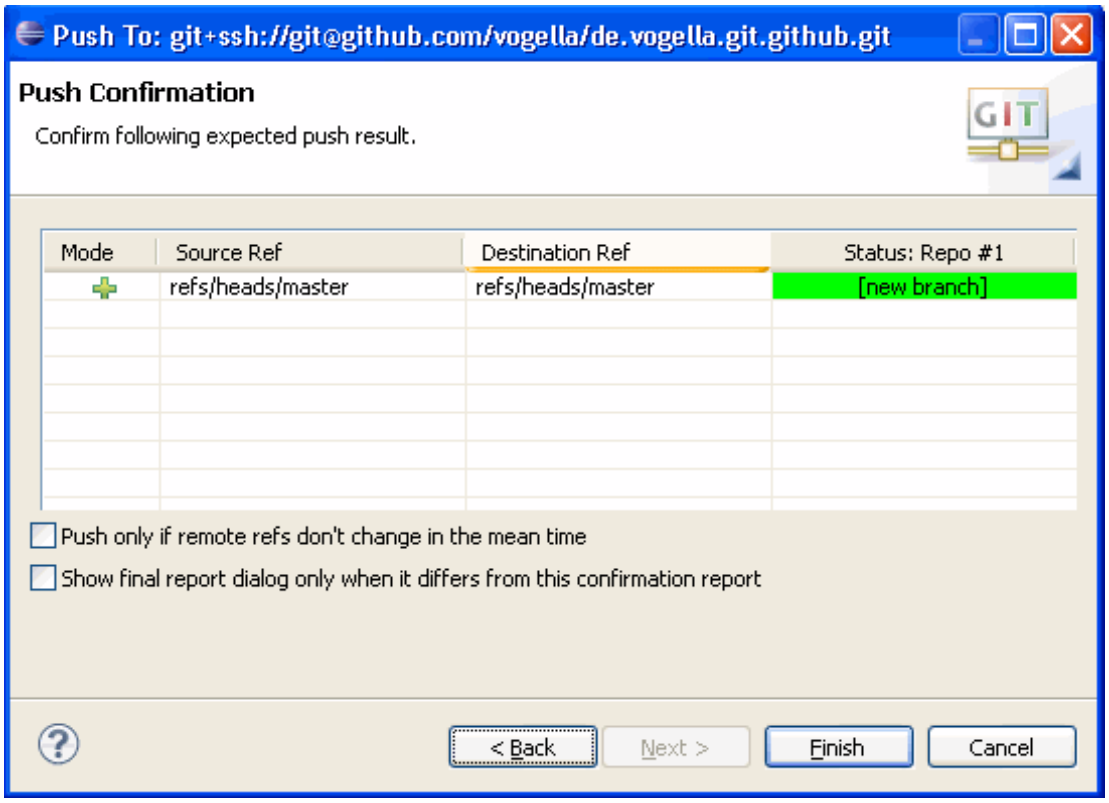
Maintain your passphase which you maintained during the Github setup.



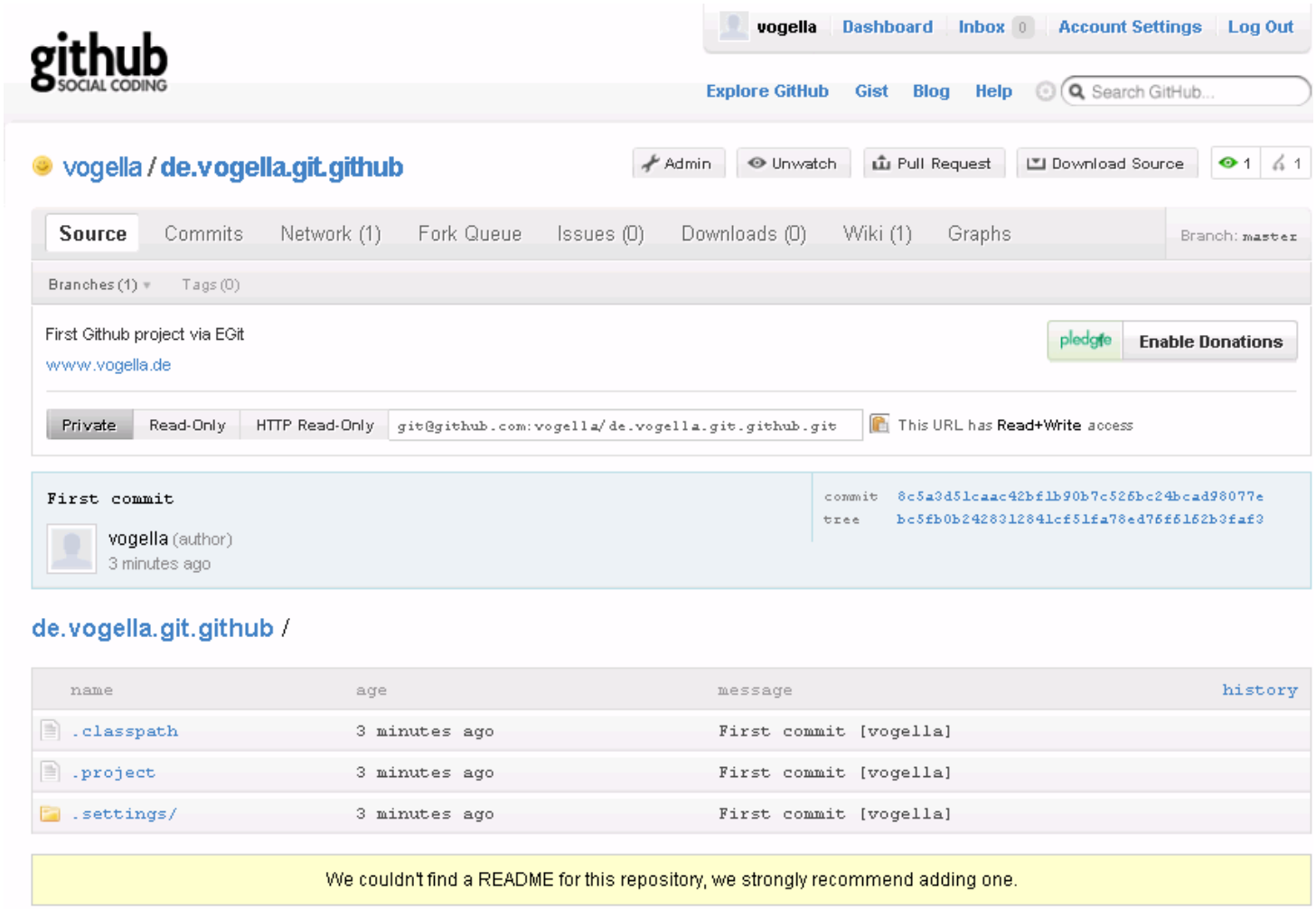
Select your branch (you should currently only have master if you followed my tutorial), press "Add all branches spec" and select next.



Press finish.



If you now select your github Dashboard and then your project you should see the commited files.



7.4. Clone repository from Github

Use the same URI you use to push to Github to clone the project into another workspace.

7.5. Mylyn integration with Github

Eclipse Mylyn is a productively tool for programmers. There is a GitHub connector for Mylyn available, please see http://wiki.eclipse.org/EGit/GitHub/UserGuide for details. .

8. Hacking EGit - Getting the source code

EGit is self-hosted on git://egit.eclipse.org. You can clone the EGit code from the repository using EGit using the following URL git://egit.eclipse.org/jgit.git and git://egit.eclipse.org/egit.git.

You also need some libraries from Orbit. See Libraries from Orbit for getting these libraries.

9. Thank you

Please help me to support this article:

10. Questions and Discussion

Before posting questions, please see the vogella FAQ. If you have questions or find an error in this article please use the www.vogella.de Google Group. I have created a short list how to create good questions which might also help you.

11. Links and Literature

11.1. EGit and Git Resources

http://www.vogella.de/articles/Eclipse/article.html EGit IDE Tutorial

http://wiki.eclipse.org/EGit/User_Guide EGit User Guide

http://wiki.eclipse.org/EGit/Contributor_Guide EGit contributor guide

Git Tutorial

11.2. vogella Resources

Eclipse RCP Training (German) Eclipse RCP Training with Lars Vogel

Android Tutorial Introduction to Android Programming

GWT Tutorial Program in Java and compile to JavaScript and HTML

Eclipse RCP Tutorial Create native applications in Java

JUnit Tutorial Test your application

Git Tutorial Put everything you have under distributed version control system