**Getting Started**

**Overview**

If you're new to Git or distributed version control systems generally, then you might want to read [Git for Eclipse Users](http://127.0.0.1:64856/help/topic/org.eclipse.egit.doc/help/EGit/Git_For_Eclipse_Users/Git-For-Eclipse-Users.html) first. More background and details can be found in the on-line book [Pro Git](http://progit.org/book/).

If you are coming from CVS, you can find common CVS workflows for Git [Platform-releng/Git Workflows](http://wiki.eclipse.org/Platform-releng/Git_Workflows).

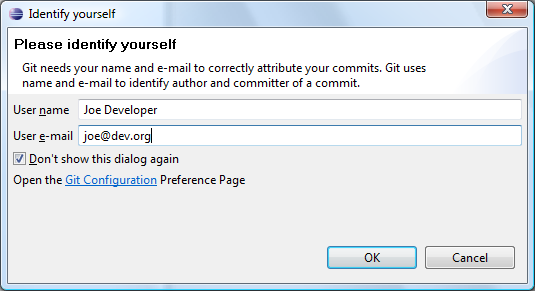
**Basic Tutorial: Adding a project to version control**

**Configuration**

***Identifying yourself***

Whenever the history of the repository is changed (technically, whenever a commit is created), Git keeps track of the user who created that commit. The identification consists of a name (typically a person's name) and an e-mail address. This information is stored in file ~/.gitconfig under dedicated keys.

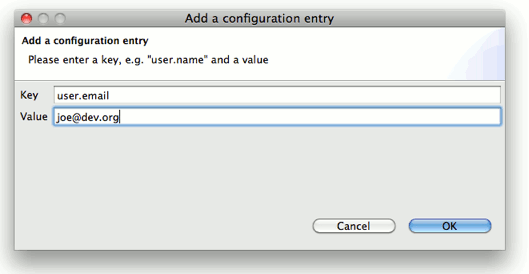
EGit will ask you for this information when you create your first commit. By default, this dialog is shown only once until you create a new workspace or tick the checkbox "Show initial configuration dialog" on the Git Preference page:

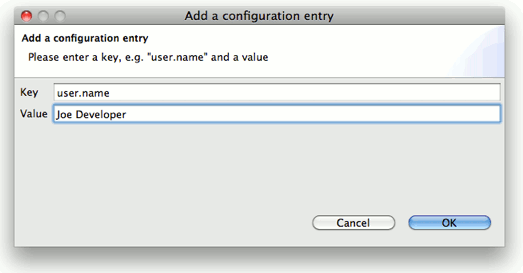


You can also untick "Don't show this dialog again" if you want to see it again later.

Instead of using this dialog, you can always change this information using the Git configuration:

* Click **Preferences > Team > Git > Configuration**
* Click **New Entry** and enter the key value pairs user.email and user.name





***Setting up the Home Directory on Windows***

Add the environment variable HOME to your environment variables.

1. In Windows 7, type "environment" at the start menu.
2. Select "Edit environment variables for your account".
3. Under "User Variables", click the "New" button.
4. Enter "HOME" in the name field.
5. Enter "%USERPROFILE%" or some other path in the value field.
6. Click OK, and OK again. You have just added the Home directory on Windows.<br />
7. File > Exit, then restart the application.

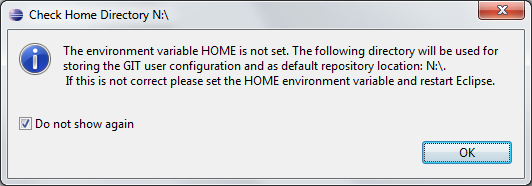
Note that if you use [Cygwin](http://www.cygwin.com/), you should already have HOME set. For example, if you installed Cygwin to C:\cygwin and your username is Fred, then Cygwin should have already set HOME to C:\cygwin\home\Fred (or something similar). You can verify this by entering echo %HOME% in the Windows command prompt, or echo $HOME in the Cygwin shell.

EGit needs this path for looking up the user configuration (.gitconfig). HOME should point to your home directory e.g. C:\Users\Tom. **Ensure correct case!** E.g. C:\users instead of C:\Users may cause problems!

If the HOME variable is not defined the home directory will be calculated by concatenating HOMEDRIVE and HOMEPATH.

If both HOME and HOMEDRIVE are not defined HOMESHARE will be used.

EGit shows a warning if HOME is not defined explicitly. Keep in mind that if you set the HOME environment variable while Eclipse is running, you will still see following warning. You will have to restart Eclipse for it to recognize the HOME value.



***Pointing out the System wide configuration***

If you use Git for Windows as a companion to EGit, make sure EGit knows where Git is installed so it can find the "system wide settings", e.g. how core.autocrlf is set. Go to the settings and look under Team>Git>Configuration and then the System Settings tab.

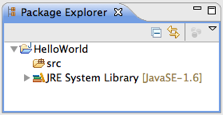
If you selected one of the options to use Git from the Command Line Prompt when you installed Git for Windows, then the location of the system wide settings is filled in with a path and everything is fine. If not, use the Browse button to locate where Git is installed, e.g. C:\Program Files(x86)\Git.

This advice also applies to users of other Git packagings, e.g. Git under Cygwin or TortoiseGit.

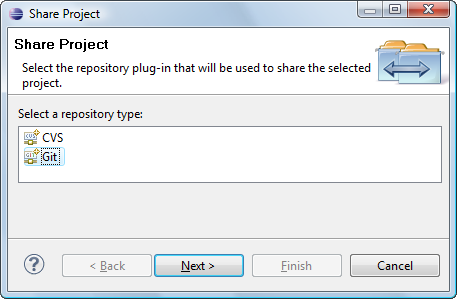
Non-Windows users should in theory check this setting, but the system wide settings are usually not used on non-Windows platforms.

**Create Repository**

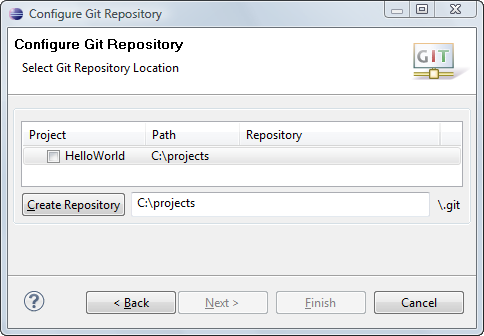
* Create a new Java project HelloWorld. (In this case, the project was built outside of your Eclipse Workspace.)



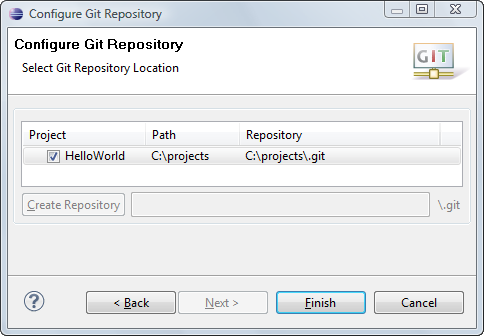
* Select the project, click **File > Team > Share Project**.
* Select repository type **Git** and click **Next**.



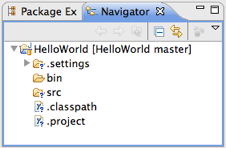
* To configure the Git repository select the new project HelloWorld.



* Click **Create Repository** to initialize a new Git repository for the HelloWorld project. If your project already resides in the working tree of an existing Git repository the repository is chosen automatically.



* Click **Finish** to close the wizard.
* The decorator text "[master]" behind the project shows that this project is tracked in a repository on the *master* branch and the question mark decorators show that the .classpath and .project and the .settings files are not yet under version control.

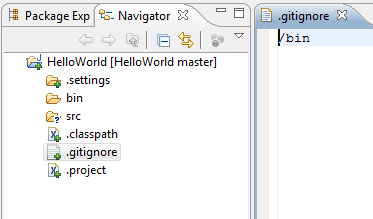


**Track Changes**

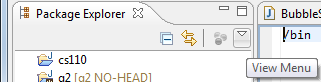
* Click **Team > Add to Index** on the project node. (This menu item is named **Add** on older versions of Egit.)
* The *+* decorators show that now the project's files have been added to version control.
* Mark the "bin" folder as "ignored by Git", either by right-clicking on it and selecting **Team > Ignore** or by creating a file .gitignore in the project folder with the following content:

/bin

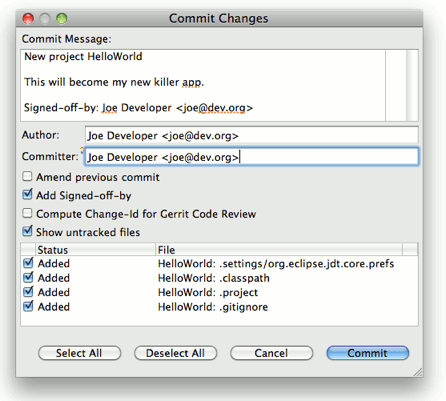
* This excludes the bin folder from Git's list of tracked files.
* Add .gitignore to version control ( **Team > Add**):



* You may have to set your Package Explorer filters in order to see .gitignore displayed in the Package Explorer. To access filters, select the down arrow on the right edge of the Package Explorer tab to display View Menu.



* Select **Filters...** from the View Menu and you will be presented with the Java Element Filters dialog. Unselect the top entry to display files that begin with . (period) such as .gitignore.
* Click **Team > Commit** in the project context menu.
* Enter a commit message explaining your change, the first line (followed by an empty line) will become the short log for this commit. By default the author and committer are taken from the .gitconfig file in your home directory.
* You may click **Add Signed-off-by** to add a *Signed-off-by:* tag.
* If you are committing the change of another author you may alter the author field to give the name and email address of the author.
* Click **Commit** to commit your first change.



* Note that the decorators of the committed files have changed as a result of your commit.

