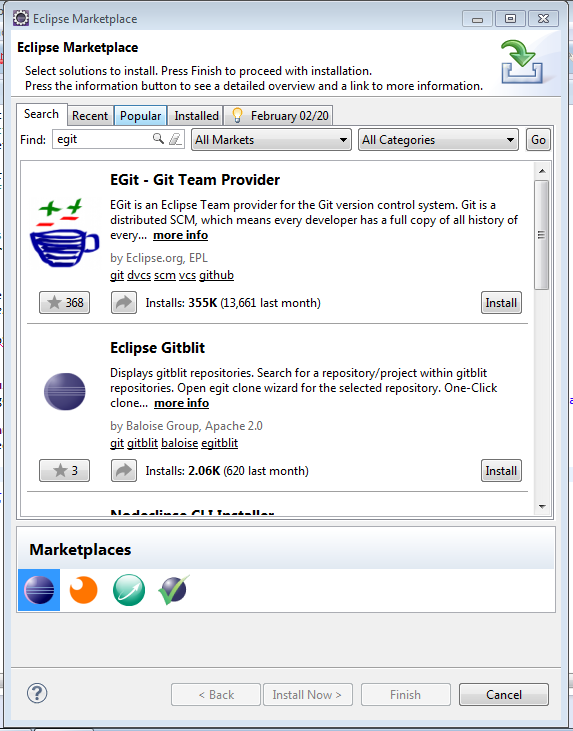
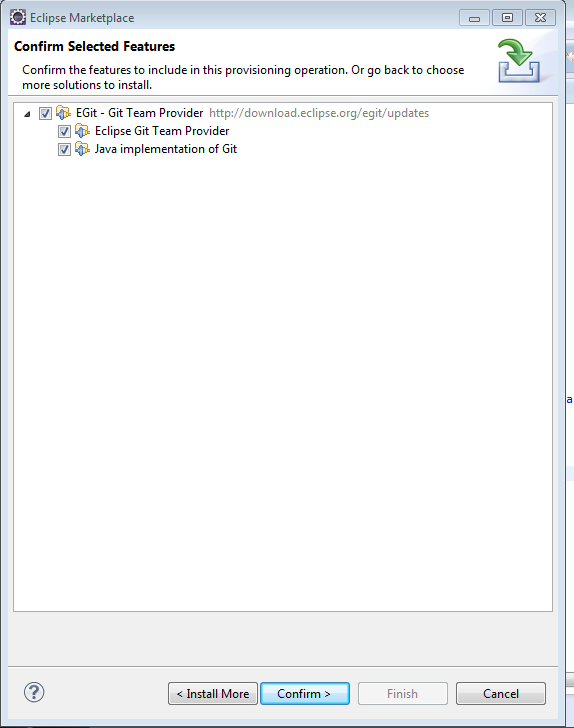
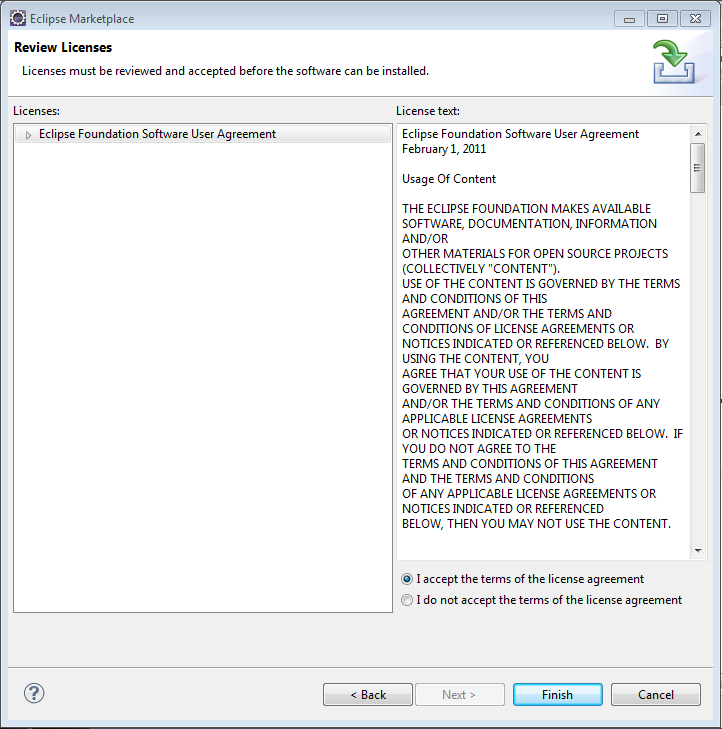
Starting with Egit

1. Steps to include Egit plugin in Eclipse IDE:





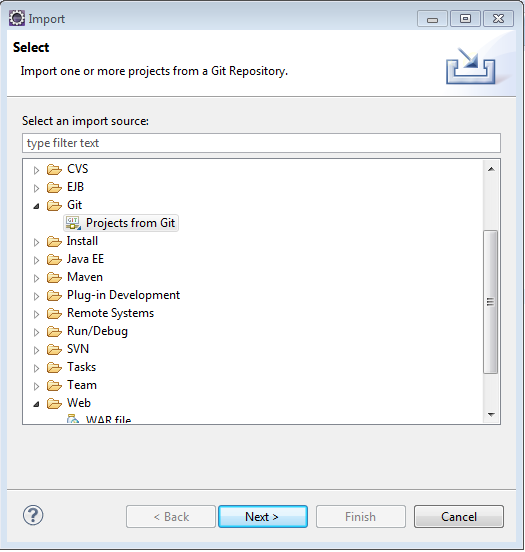
Accept and install



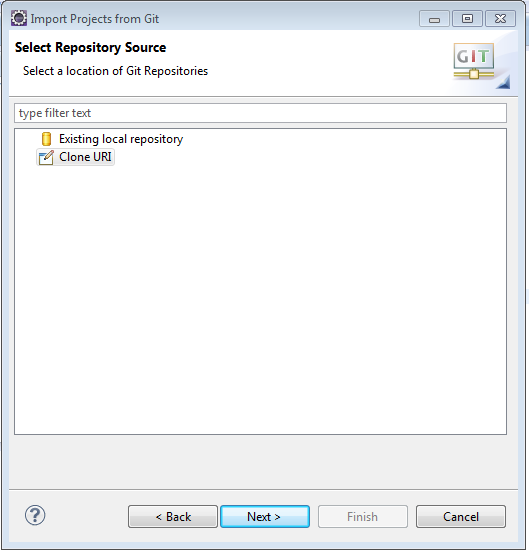
Restart your eclipse after installation is complete for changes to take effect.

**Import a Git Project into workspace:**

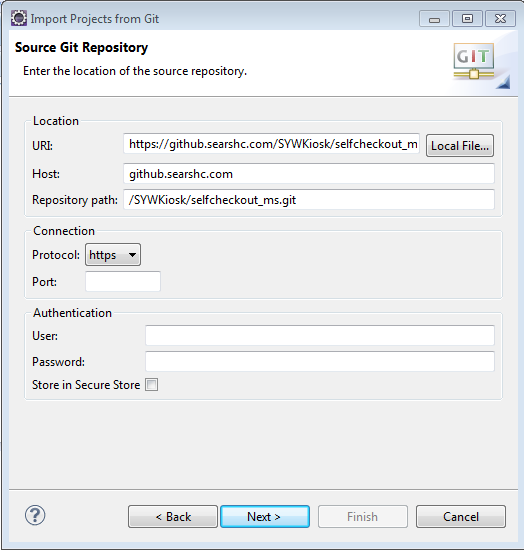
1. Under eclipse File Menu click on Import. Select Git 🡪 Projects from Git and click Next.



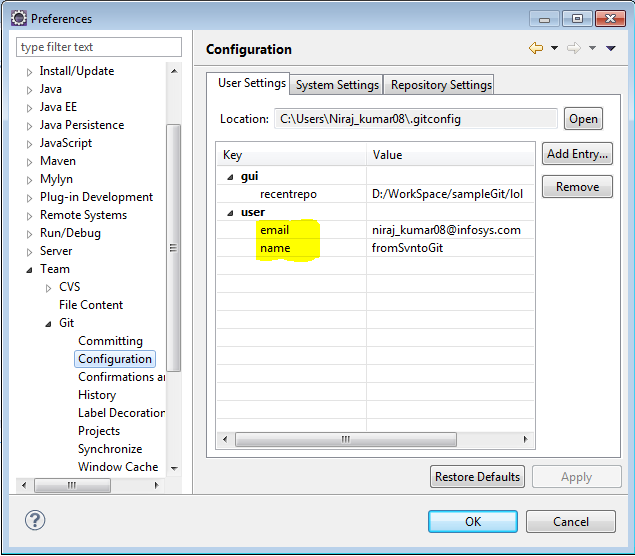
1. On Next screen, select CloneURI to clone directly from github else select existing local repository to add a local repository to workspace.



1. In the next screen, enter URI of the repository and click next.



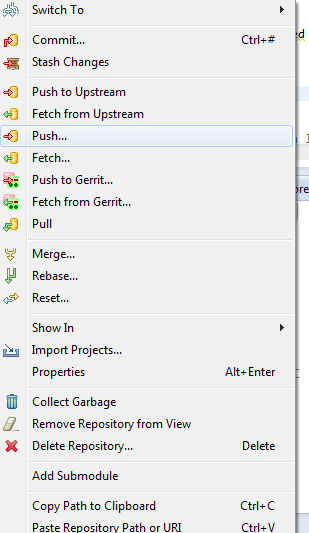
1. On Branch Selection screen, select the branch which needs to be checked out.
2. Choose a local directory to save this repository in. By default the value is “C:\Users\UserName\git\ProjectName” and click Next.
3. Select the working directory and click Next.
4. Select the Projects in import projects screen and click Finish.
5. Configure your username and e-mail in preferences for git. The username and email should be the same you use for your Git account, ie. your GitHub account.



**Committing the changes:**

In order to commit the code changes, right click on project and go to Team🡪Commit. Enter the commit message and select the files to be committed and click on **commit** at bottom of page.

The change will be committed to repository only after **push** operation is performed.



**Stashing local changes:**

Stashing in Git refers to taking all changes made in your working directory and committing them to a separate area known as the “stash”. It is a great way to save the current uncommitted changes without creating a new commit on the current branch. Once the stashing is complete, the working directory and index are reset to the current branch’s head revision.

Stashing is common when you want to merge or rebase but don’t want to lose your local uncommitted changes but you aren’t quite ready to commit them on the current branch.  You can first stash your work, perform the merge or rebase and then apply the stash to get the changes back in your working directory.

1. **Stashing Local changes** :

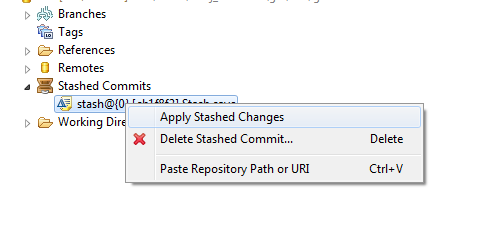
* Open the Git Repositories view
* Select the Stash Changes option from the context menu available on the project repository.
* Enter the stashed message (optional).

1. **Browsing stashed changes**:

* Open the Git Repositories view.
* Expand “**Stashed Commits”** node to see all the changes which are stashed.

1. **Apply stashed changes to current code**:

* After finding the appropriate stashed change, right click on the **Stashed Commits** node.
* Select **Apply Stashed Changes** option from the context menu



**Merging using EGit:**

EGit supports merging of branches to add the changes of one branch into another. In order to merge, go to project and Team →Merge to start the merge dialog.

**Solving merge conflicts-**

On getting conflicts while merging, EGit highlights the affected files. It also supports the resolution of these merge conflicts.

* Right-click on a file with merge conflicts and select Team → Merge Tool.
* On the next screen, select **“Use HEAD (the last local version) of conflicting files”** as merge mode. This way one can see the original changes on the left side and the conflicting changes on the right side.
* Edit the text on the left side or use the Copy current change from right to left button to copy the conflicting changes from right to left.
* Once the merge is complete, select Team → Add from the context menu of the resource to mark the conflicts as resolved and commit the merge resolution via Team → Commit.