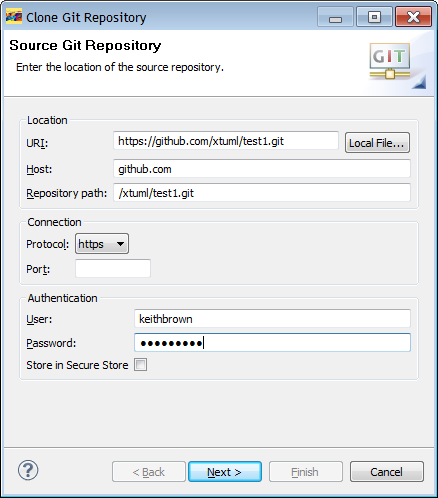
**HOWTO Use the xtUML GitHub Repository with Eclipse**

Setup

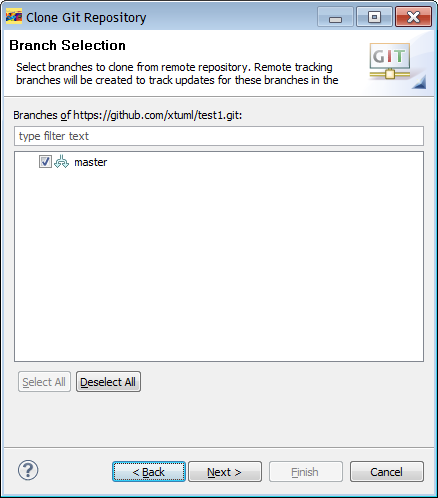
1. Open the file system explorer
2. Create a folder to hold your git repositories (e.g. C:/git/)
3. Install BridgePoint
4. Start BridgePoint
5. Open the Git Repository Exploring Perspective

Clone a Repository

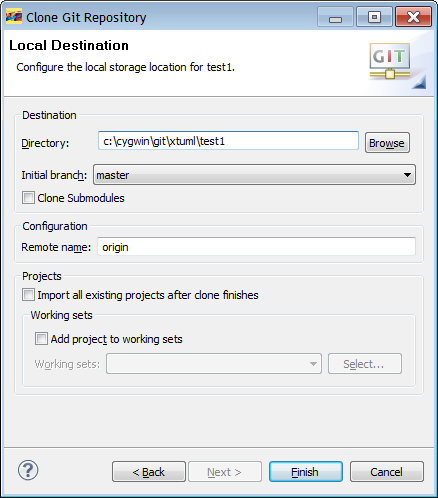
1. Clone the repository (here using example repository xtuml/test1)
   1. Click the "Clone a Git repository" link
   2. Enter the appropriate information as shown here:



* 1. Click **Next**
  2. In the Branch Selection dialog, leave the default selection "master" checked. If other branches are present, leave them checked as well.



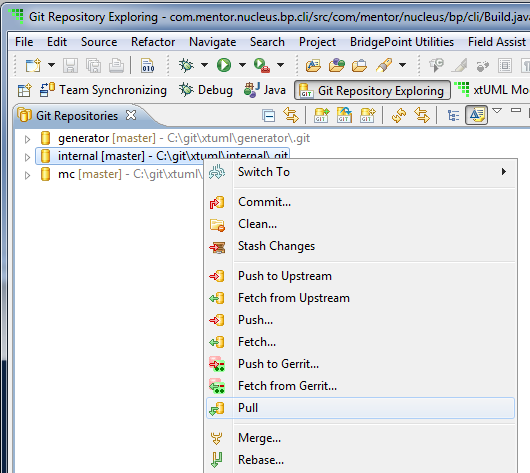
* 1. Click **Next**
  2. In the Local Destination dialog, change the directory to reside under the git repositories directory created above (e.g. c:\git\xtuml\test1 ). **Do not** clone the repository to a destination under the workspace!
     1. Leave "Clone Submodules" unchecked.
     2. Chose if you want to import all projects now, or leave unchecked to only import select ones after cloning.



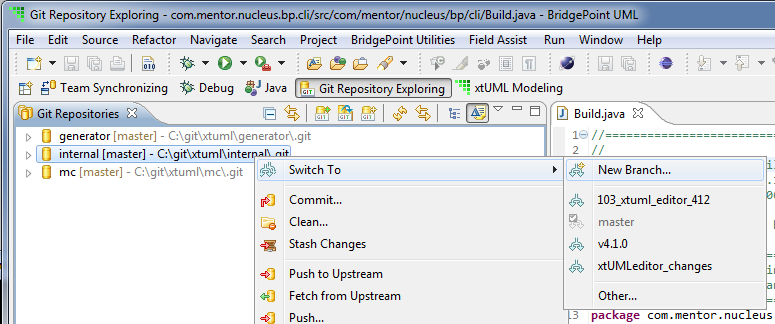
* 1. Click **Finish**

Creating a Branch

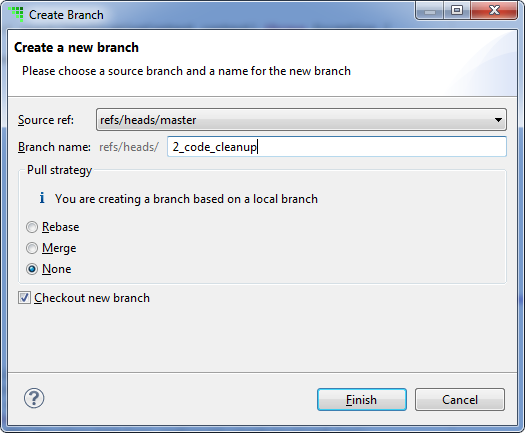
1. Switch to master and Pull to make sure you are starting for the latest code base.



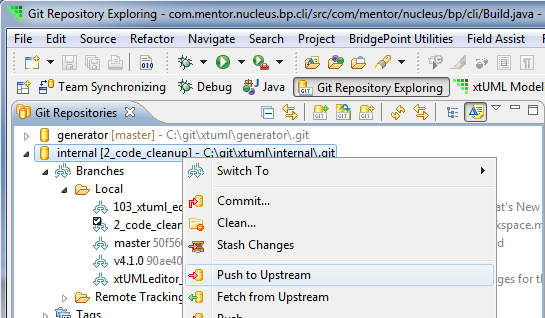
1. Right-click on the repository and select **Switch To > New Branch…**



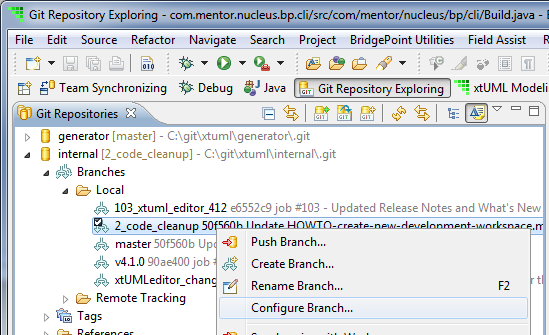
1. In the Create Branch dialog, enter the desired branch name (e.g. <git issue#>\_<short description>)



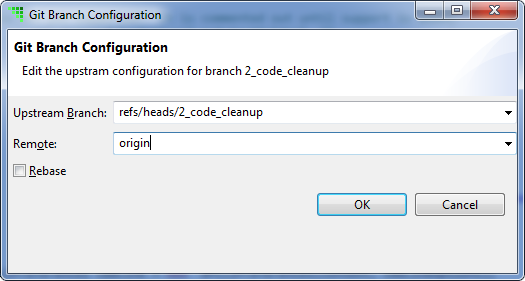
1. The new branch is created and the branch name shows next to the repository. Next, right-click on the repository and select **Push to Upstream…** to upload the new branch to github



1. Expand the **<Repository> > Branches > Local**. Right-click on the branch and select **Configure Branch…**



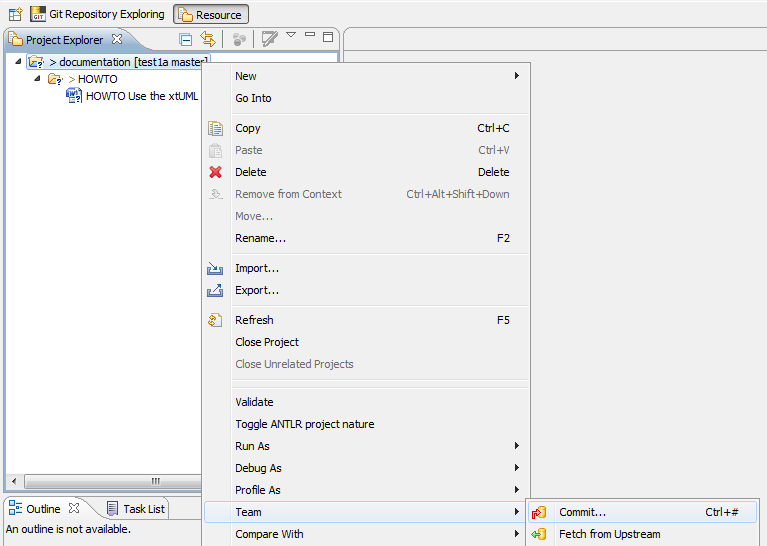
1. Enter the information:
   1. Upstream branch: refs/heads/<your branch name>
   2. Remote: origin



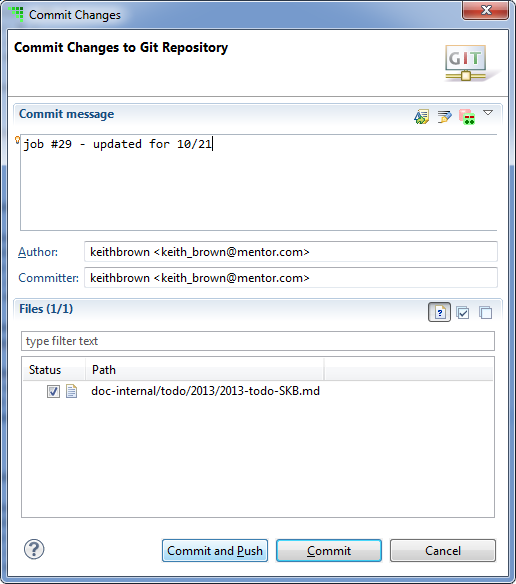
1. You are now ready to begin development, make changes, and commit and push the changes.

Committing Changes

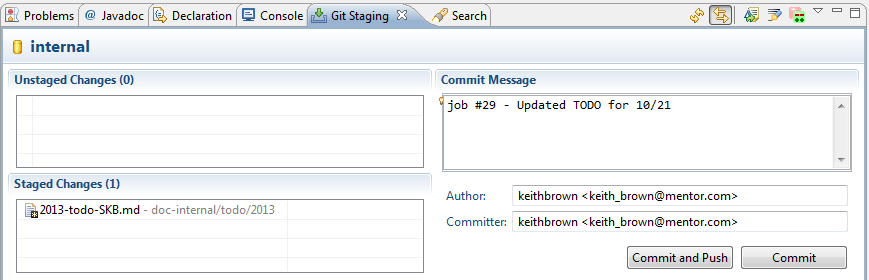
1. The files now show as dirty in the workspace. In the context menu, select **Team > Commit…**



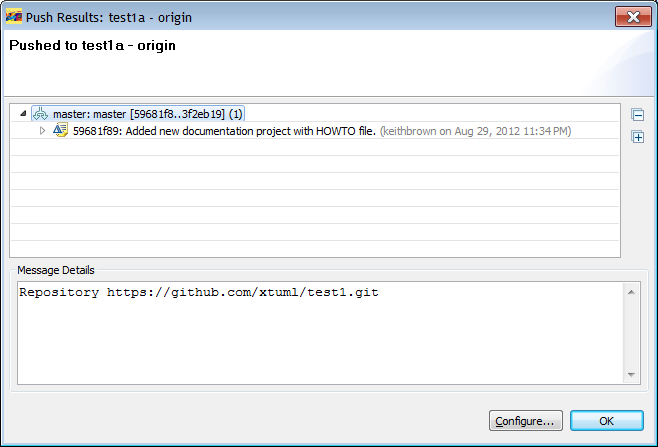
1. Enter the commit comment and select the files to commit. The commit comment must start with “job #<git issue>” to tie the commit into the issue tracker.



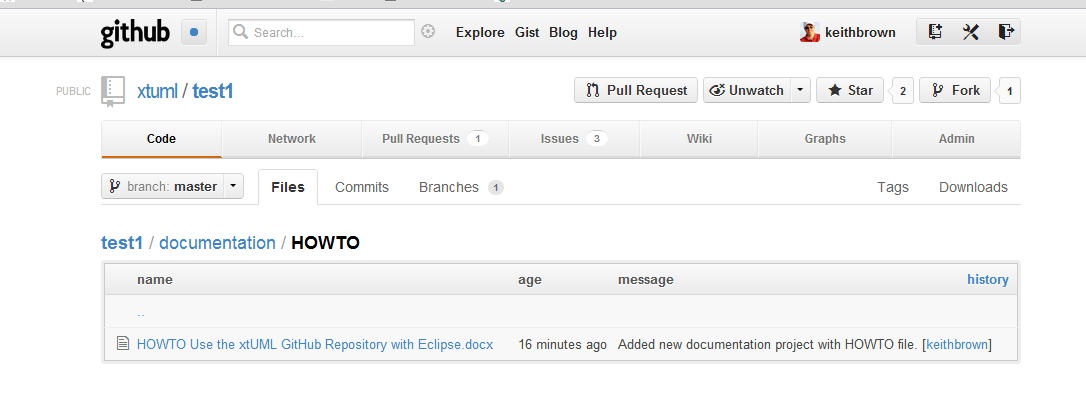
1. Click **Commit and Push**
2. Alternatively, you can use the Git Staging View. Drag and drop files from the Unstaged Changes list to the Staged Changes list. Then enter a Commit Message and click **Commit and Push**.



1. When the push is complete you will get a result message

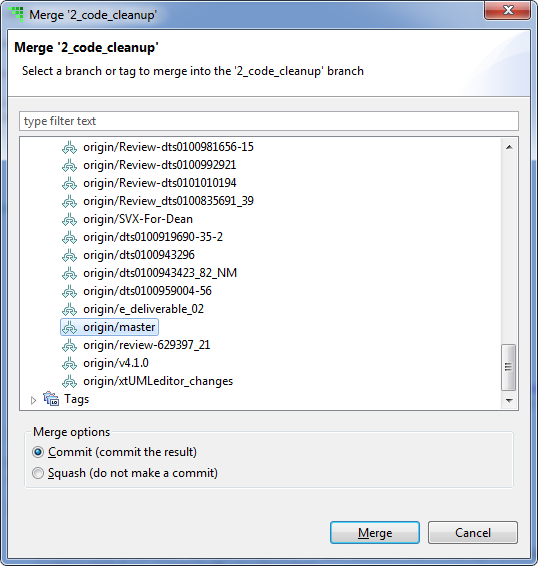


1. After the push is complete, the changes (here, new data) show up on github.com



Getting Up to Date with Master – Preparing to Create a Pull Request

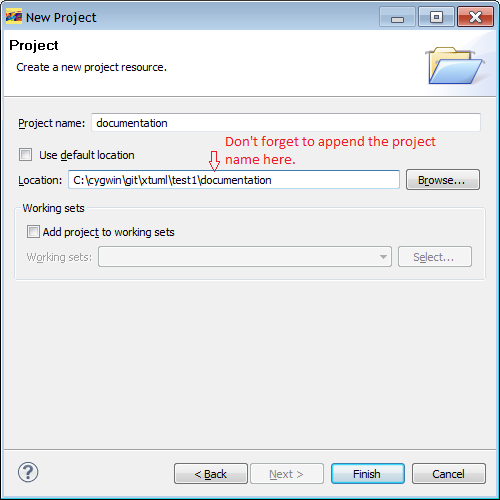
1. Switch to Git Repository Exploring perspective
2. Right-click on the repository and select **Pull**
3. Expand the Local and Remote Tracking branches under the repository
4. Use **Ctrl-<click>** to select your branch from under Local Branches and origin/master from under Remote Tracking branches.
5. Right-click on either one and choose **Synchronize with each other**
6. Go through the changes and make sure you are happy with the differences
7. Right-click on the repository again and select **Merge…**
8. Choose origin/master from the list of Remote Tracking branches



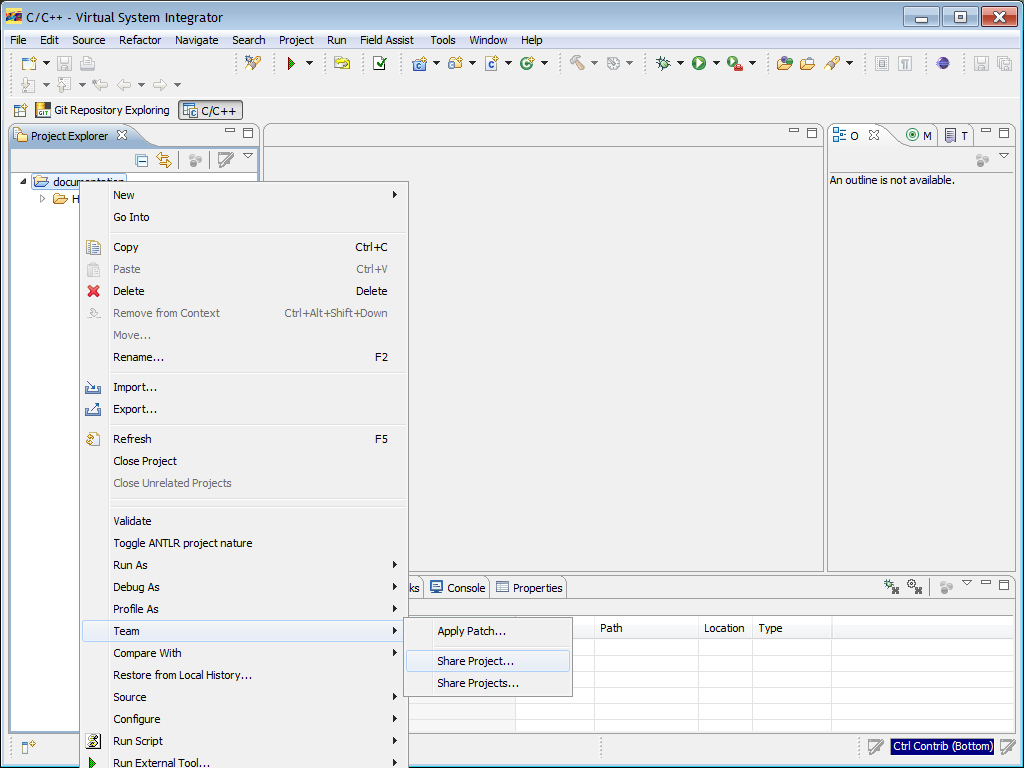
1. Resolve any merge conflicts
2. Right-click on the repository and select **Push to Upstream**
3. Your branch is now up to date with master. You can continue development or ready a code review or pull request.

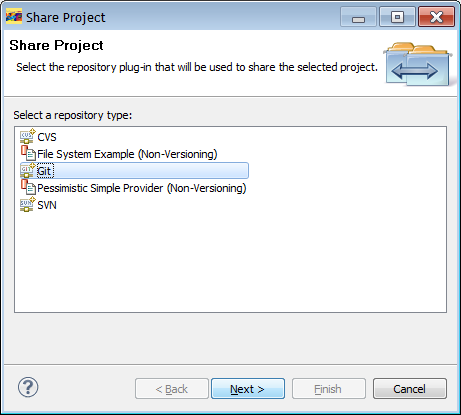
Add a New Project

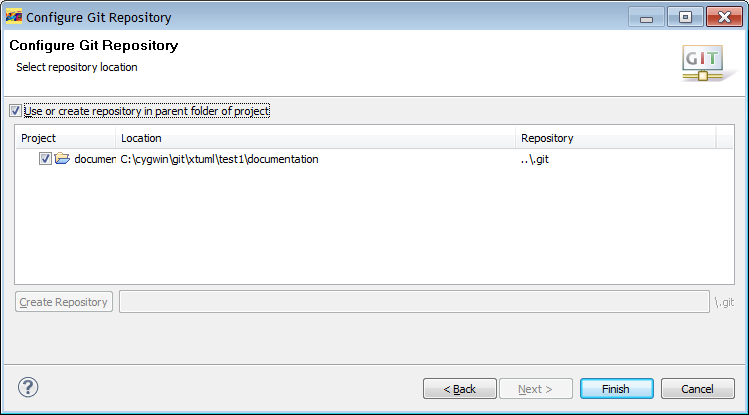
1. Create a new project
   1. Open the New Project Wizard
   2. Select the project type
   3. Click **Next**
   4. Enter the Project name and storage location. Store it to the git folder under cygwin created earlier.



* 1. Click **Finish**
  2. Add project content
  3. Now Share the project into git







* 1. Click **Finish**