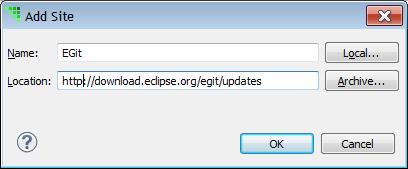
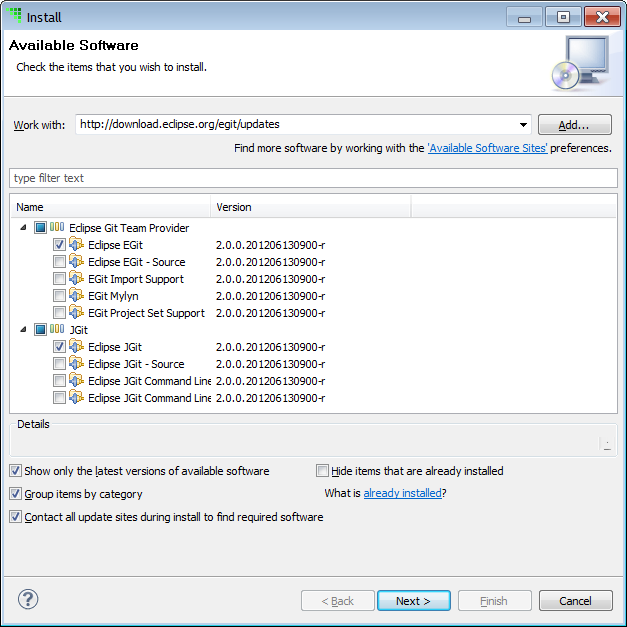
**HOWTO Use the xtUML GitHub Repository with Eclipse**

Setup

1. Install Cygwin. NOTE: this step is not required but is recommended
2. Create an account on github.com. Notify Cort you want to be added to the xtuml group
3. Prepare the data storage
   1. Open cygwin
   2. cd /
   3. mkdir git
4. Open eclipse (VSx\_Dev or BridgePoint)
   1. **Help > Install New Software...**
   2. **Add...** new site to work with:

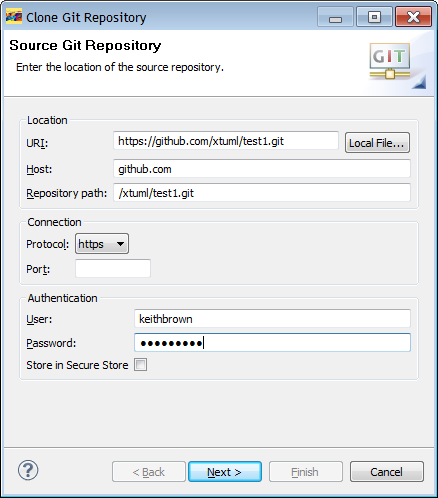


* 1. Choose only "Eclipse EGit" and "Eclipse JGit" plug-ins:

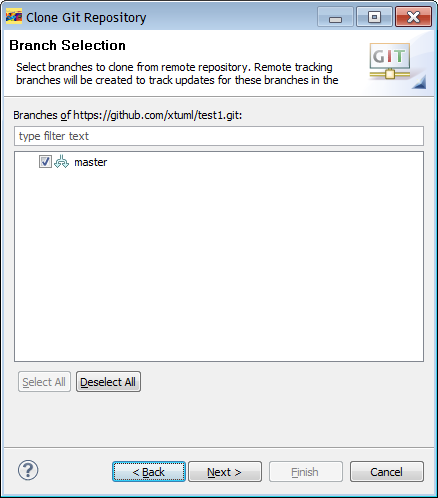


Clone a Repository

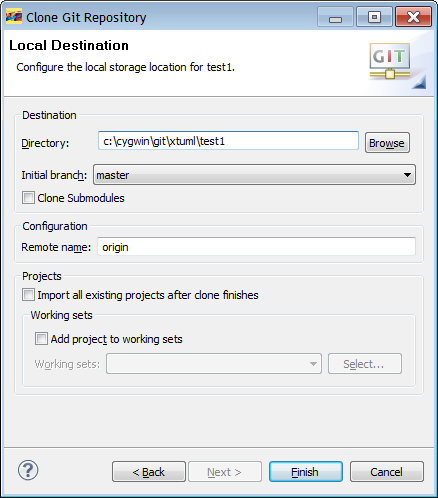
1. Clone the repository (here using example repository xtuml/test1)
   1. Open the "Git Repository Exploring" perspective
   2. Click the "Clone a Git repository" link
   3. Enter the appropriate information as shown here:



* 1. Click **Next**
  2. In the Branch Selection dialog, leave the default selection "master" checked.



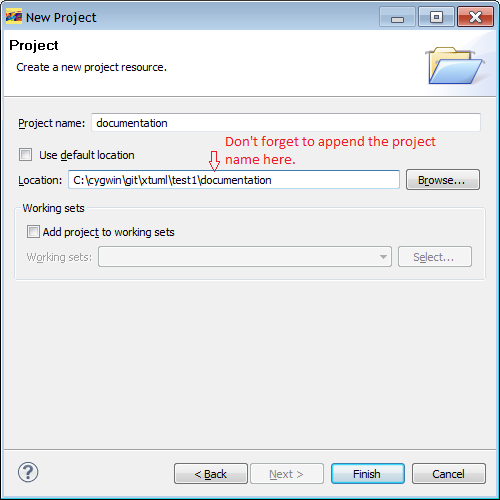
* 1. Click **Next**
  2. In the Local Destination dialog, change the directory to c:\cygwin\git\xtuml\test1
     1. Leave "Clone Submodules" unchecked.
     2. Chose if you want to import all projects now, or leave unchecked to only import select ones after cloning.



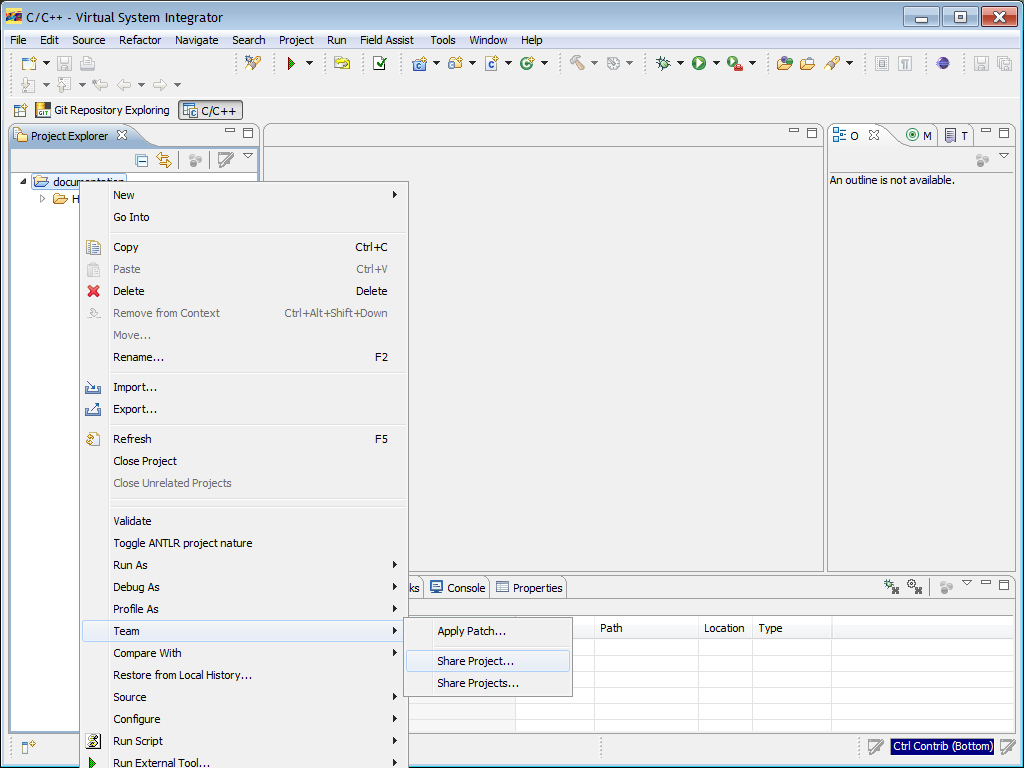
* 1. Click **Finish**

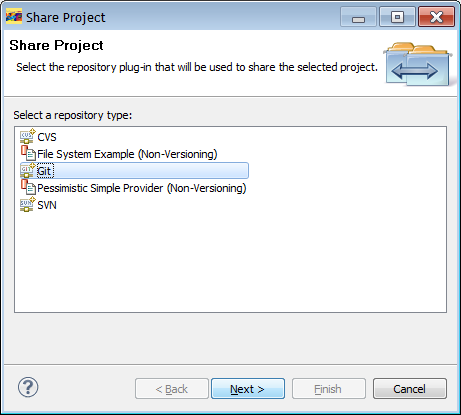
Add a New Project

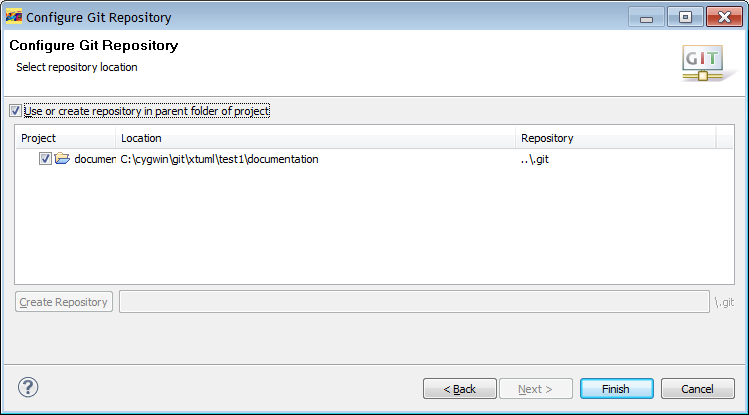
1. Create a new project
   1. Open the New Project Wizard
   2. Select the project type
   3. Click **Next**
   4. Enter the Project name and storage location. Store it to the git folder under cygwin created earlier.



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



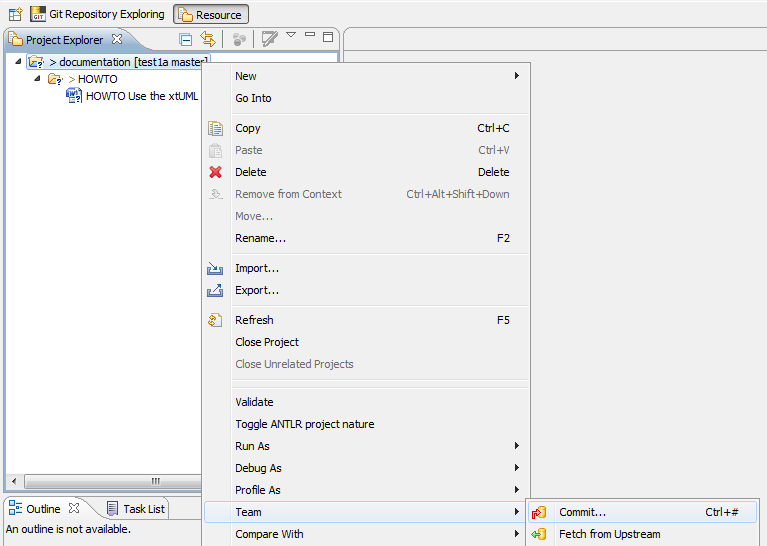




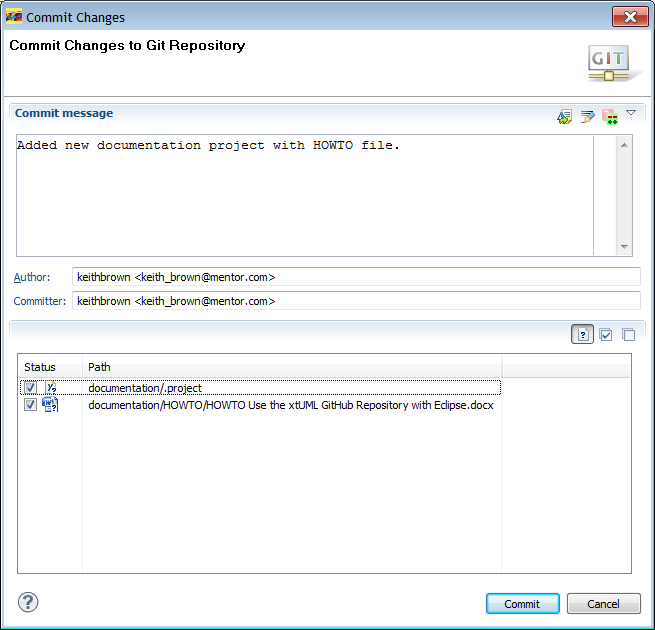
* 1. Click **Finish**

Commit the Changes

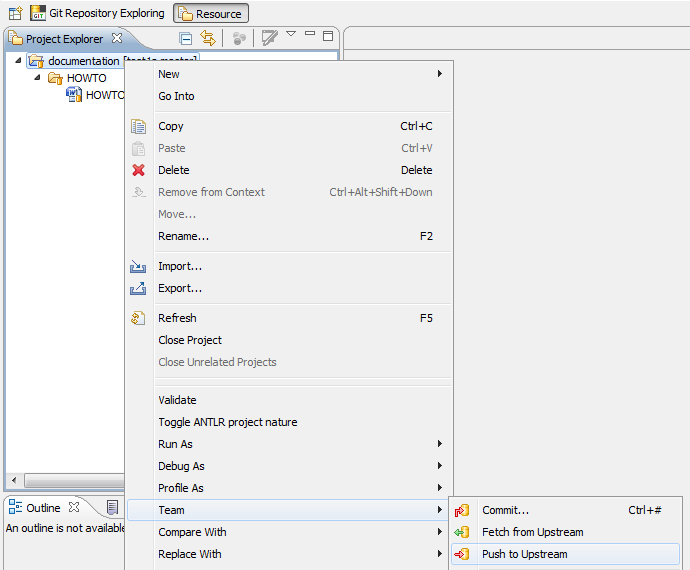
1. The files now show as dirty in the workspace. In the context menu, select **Team > Commit…**



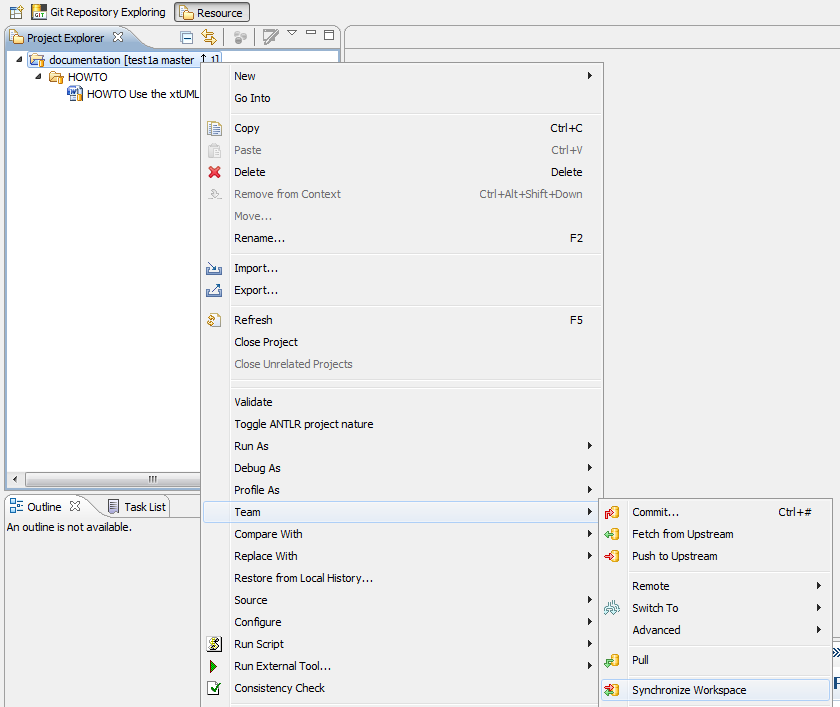
1. Enter the commit comment and select the files to commit.

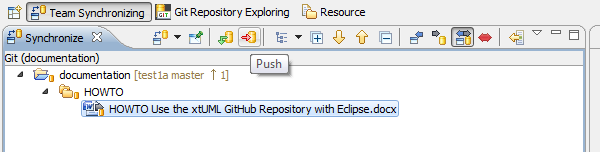


1. Click **Commit**
2. The changes are now committed to the local repository. To share the changes with the remote repository, select **Team > Push to Upstream**

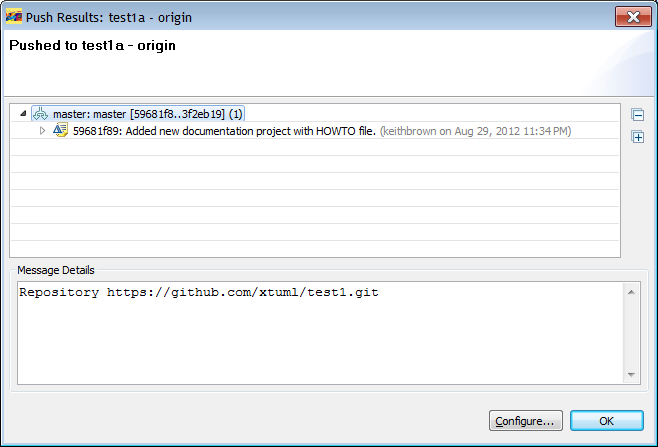


1. Alternatively, you can use the Team Synchronization tool





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

