T3000 Building Automation System

Get Started with GitHub

Temco Controls

Jay

Table of Contents

1. Introduction to Git
2. Introduction to GitHub
3. Working with GitHub
4. T3000 Building Automation System
5. How to compile
6. A typical cycle
7. SOP for development from developer’s point of view

# Introduction to Git

While developing software where developers are spread across the globe, issues of version management become critical. What if two developers simultaneously work on the same module and try to upload their changes back? Situations like these are not uncommon in real-world. To solve problems such as these there exist Version Control Systems.

## What is Git?

Git is a distributed version control system (dvcs) written in the programming language C. A distributed version control system keeps track of software revisions and allows many developers to work on a given project without necessarily being connected to a common network.

There is no central server which stores the data. Every local copy contains full history of source code.

Git keeps track of all versions. Therefore, you can revert to any point in your source code history.

The logo of git looks like below:



Git is a [distributed revision control](http://en.wikipedia.org/wiki/Distributed_revision_control) and [source code management](http://en.wikipedia.org/wiki/Source_code_management) (SCM) system with an emphasis on speed. Git was initially designed and developed by [Linus Torvalds](http://en.wikipedia.org/wiki/Linus_Torvalds) for [Linux kernel](http://en.wikipedia.org/wiki/Linux_kernel) development; it has since been adopted by many other projects. Every Git [working directory](http://en.wikipedia.org/wiki/Working_directory) is a full-fledged [repository](http://en.wikipedia.org/wiki/Repository_(version_control)) with complete history and full revision tracking capabilities, not dependent on network access or a central server. Git is [free software](http://en.wikipedia.org/wiki/Free_software) distributed under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) version 2.

## Local and Remote Repositories

In a distributed version control system everyone has a complete copy of the source code (including the complete history of the source code) and can perform version control operations against this local copy. The use of a dvcs does not require a central code repository.

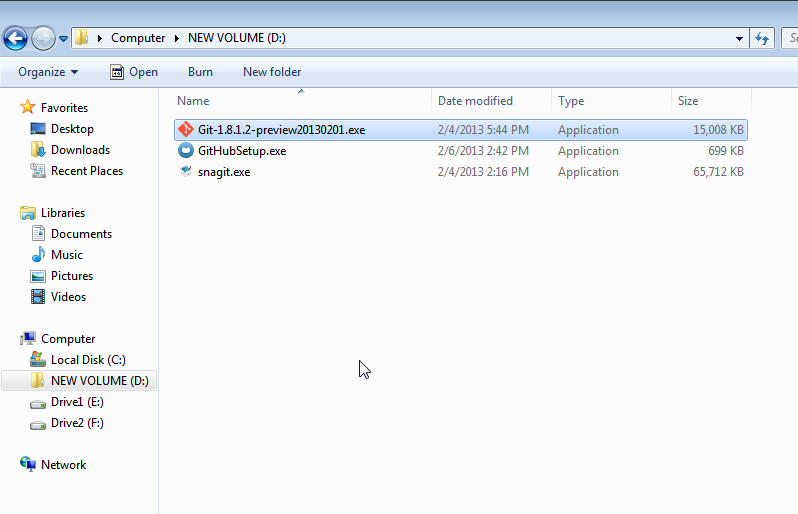
Git commits file changes to your local repository and you can synchronize your repository with other (remote) repositories. Git allows you to clone repositories, e.g. create an exact copy of a repository including the complete history of the source code. Owners of repositories can synchronize changes via push (transferring changes to a remote repository) or pull (getting changes from a remote repository).

Refer to <http://git-scm.com/book> for detailed help on understanding git.

# Installing Tools

# Installation of git

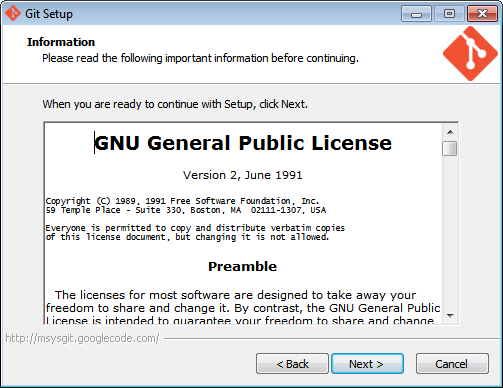
1. Download git installer.
2. Double click on the setup file.



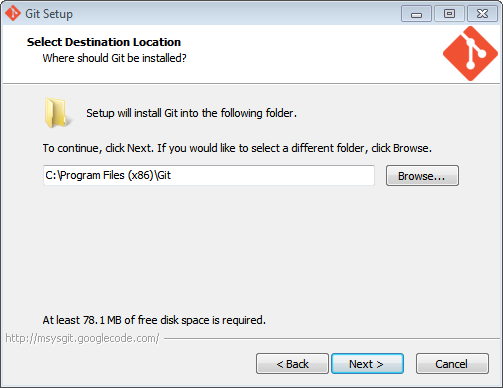
1. Following dialog box opens. Click Next to continue.



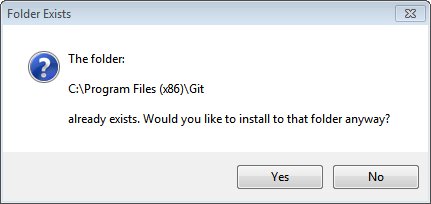
1. Read GNU General Public License. Click Next.



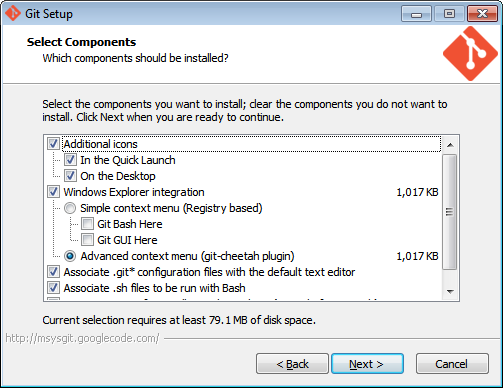
1. Select destination location. This is the directory where you want to install git. Normally, default directory can be kept as it is. If you wish, you can change the directory by clicking on Browse button. Here, we will install git in the default directory. (C:\Program Files (x86)\Git). Click Next.



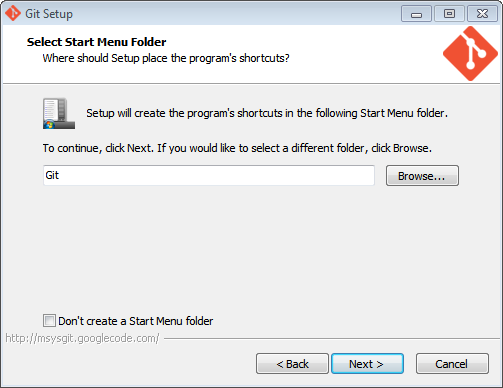
1. If following dialog box appears, click Yes.



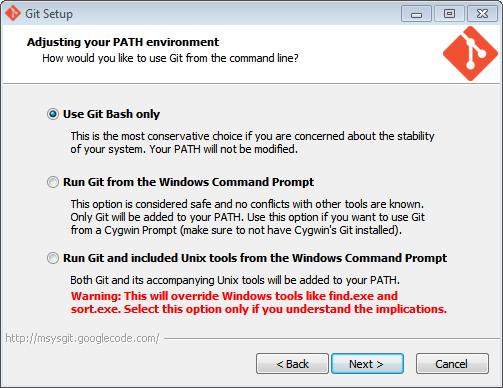
1. Following screen appears. Leave all the default options as is and click Next.



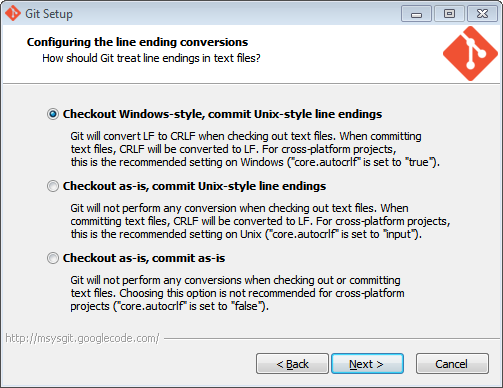
1. Following screen appears. Click Next.



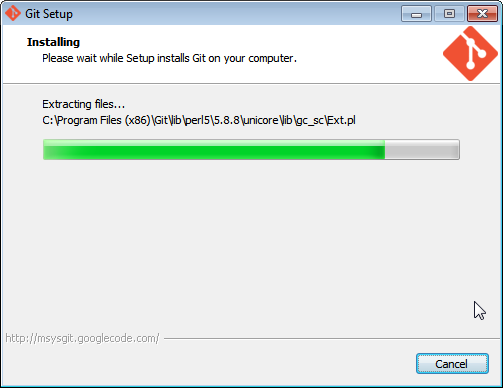
1. Following screen appears. Click Next to continue.



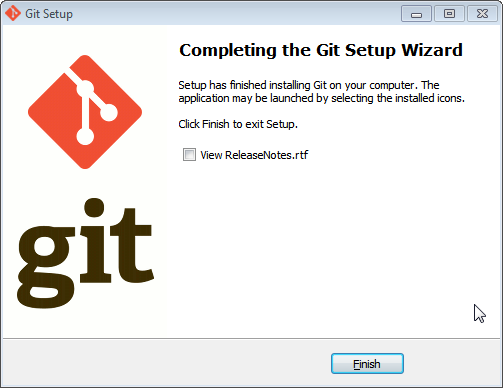
1. Following screen appears. Click Next to continue.



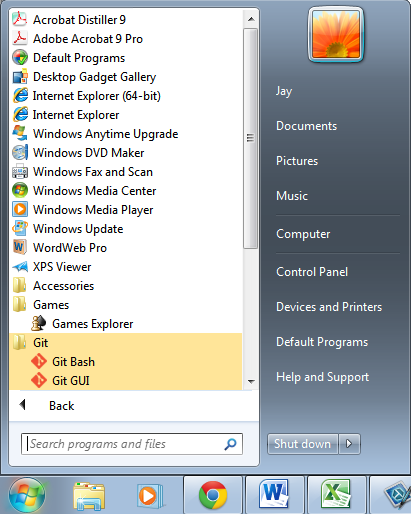
1. Setup will progress.



1. Click Finish to complete the setup.



1. Check installation in Start Menu.

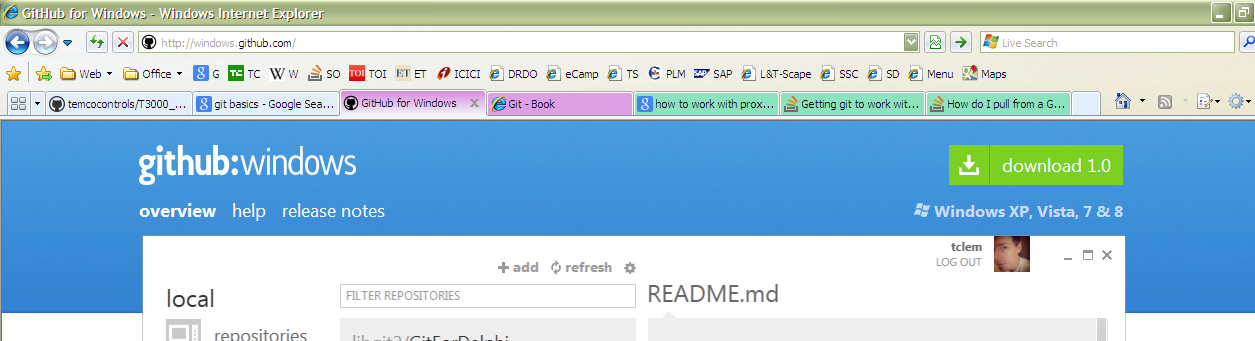


As can be seen, there are two sub menus under Git menu.

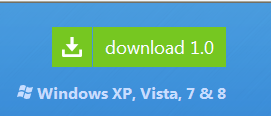
1. Git Bash
2. Git GUI

# Installation of github for windows

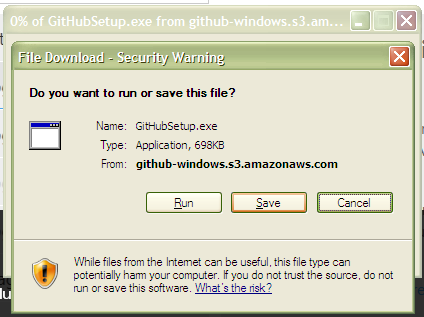
1. Download github for windows from http://windows.github.com/



1. Click on the link given for download.

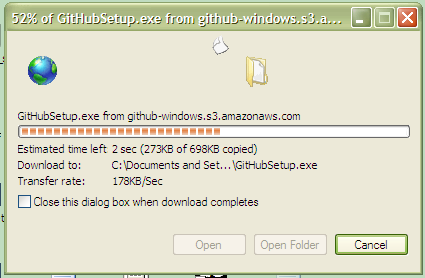


1. If any confirmation is asked by OS, click Save.



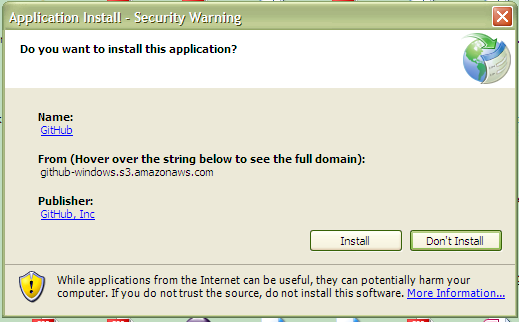
Save the file on your computer at a chosen location.

1. File download progress bar would be shown.

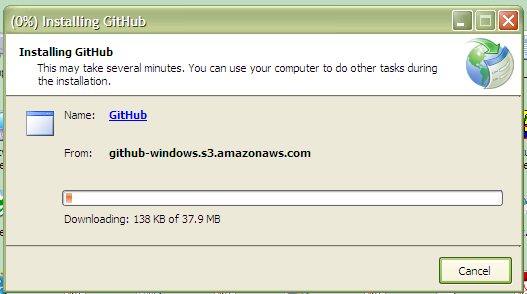


Once the download is complete, double-click on the same to install github for windows on your PC.

1. A dialog box for Security warning may appear.

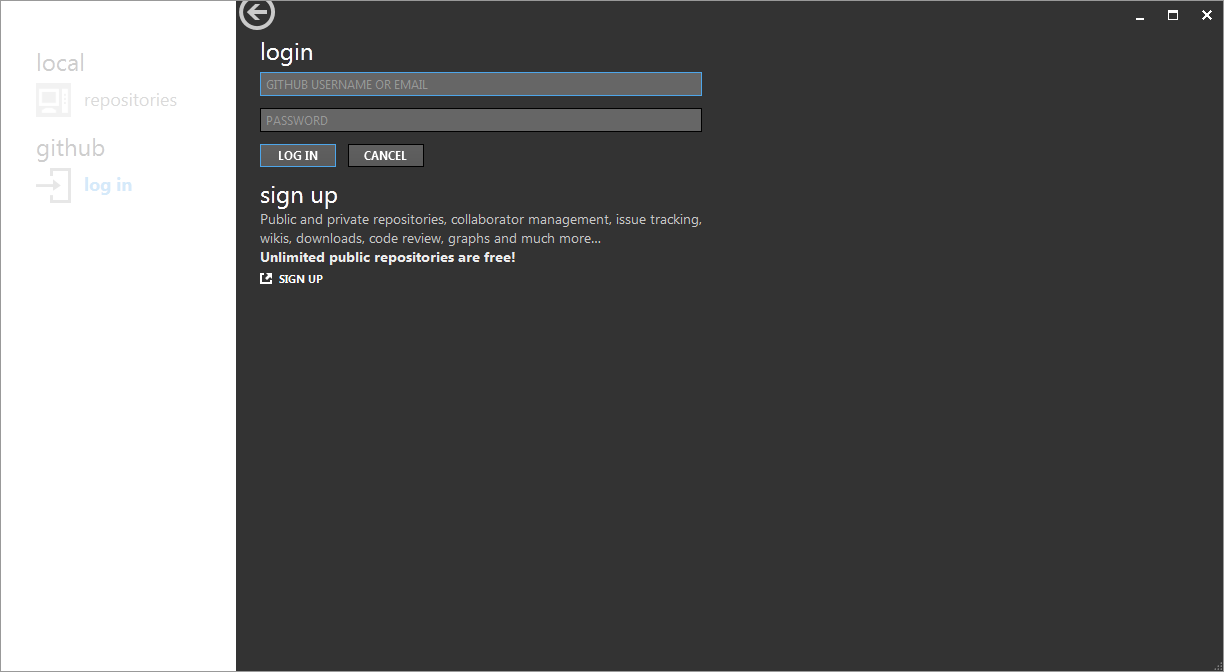


1. Github for windows starts installing.

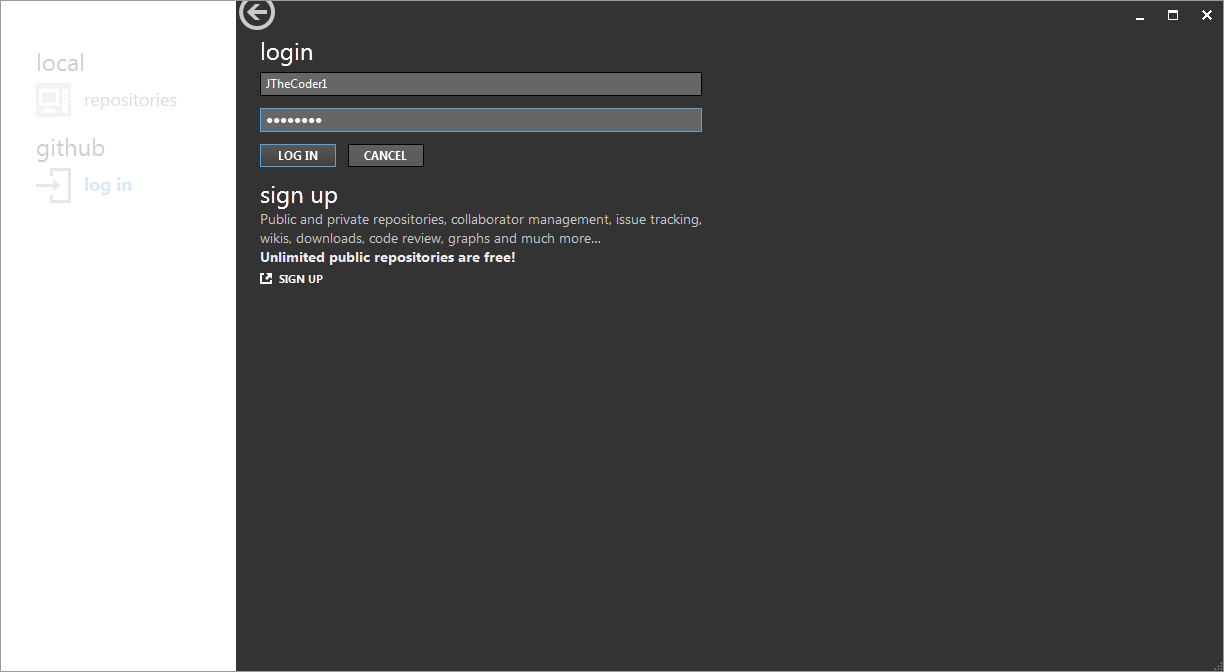


Istallation eventually gets over. Application starts.

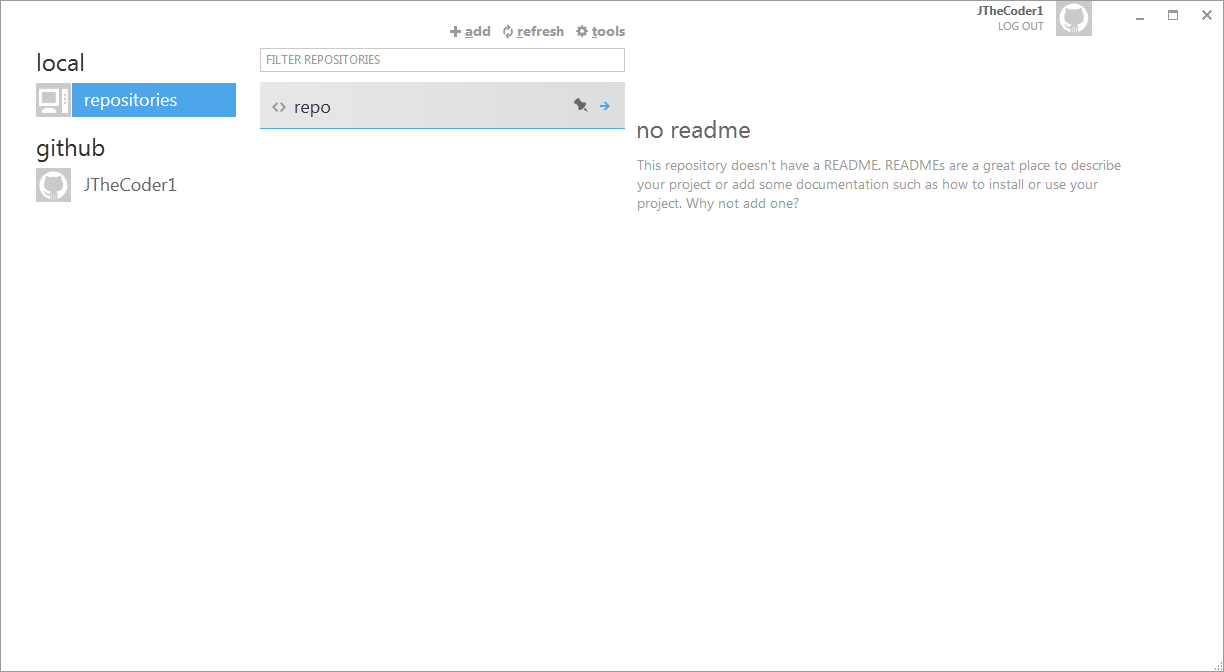
1. The first screen in the application Github for Windows is as follows.



1. Enter username and password in the given fields. You have to Sign Up with github to have these information.

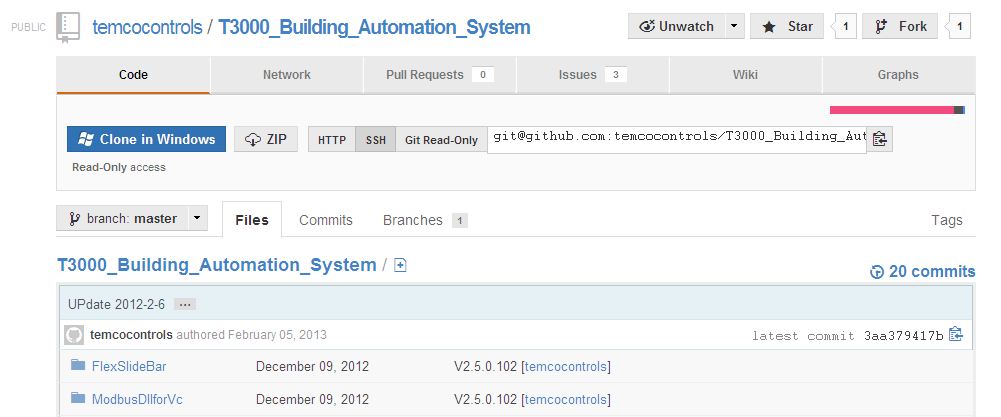


1. Application screen looks as follows.

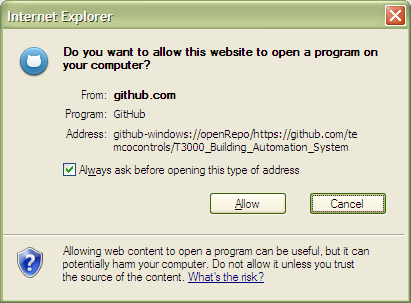


1. Once the application is installed on the PC, go to github website. Open the project you want to work on. For example, in my case, I am opening repository of T3000 Building Automation System.

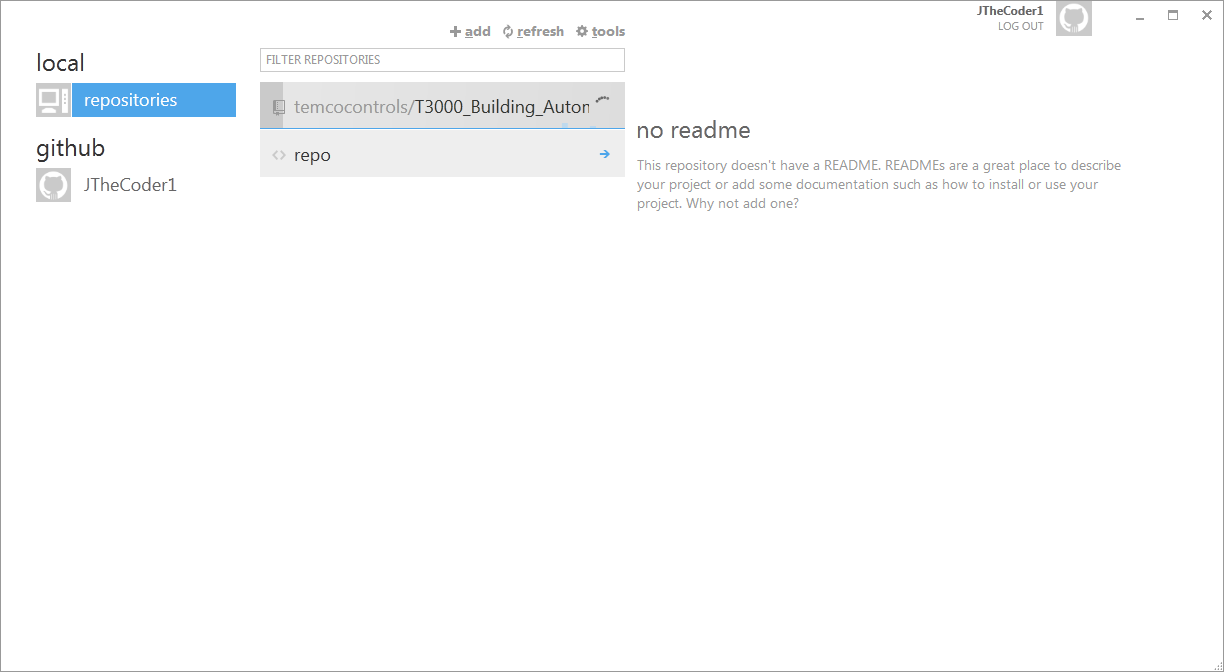
In order to clone the repository from github to local PC, click on **Clone in Windows** button as shown below.



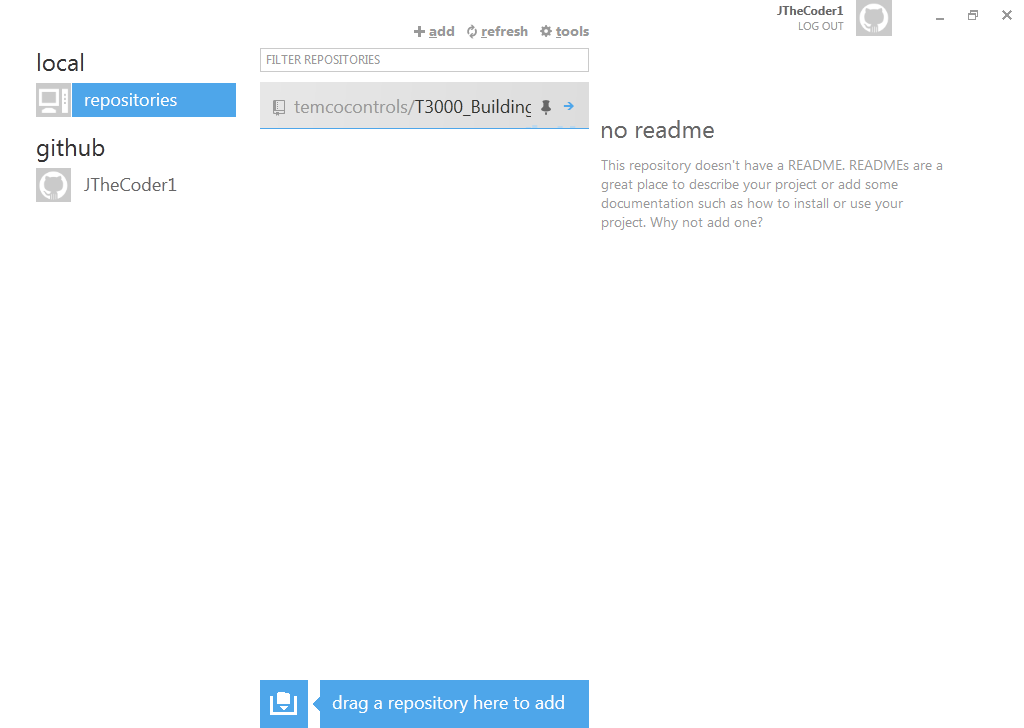
1. You may get following warning message. Click Allow.



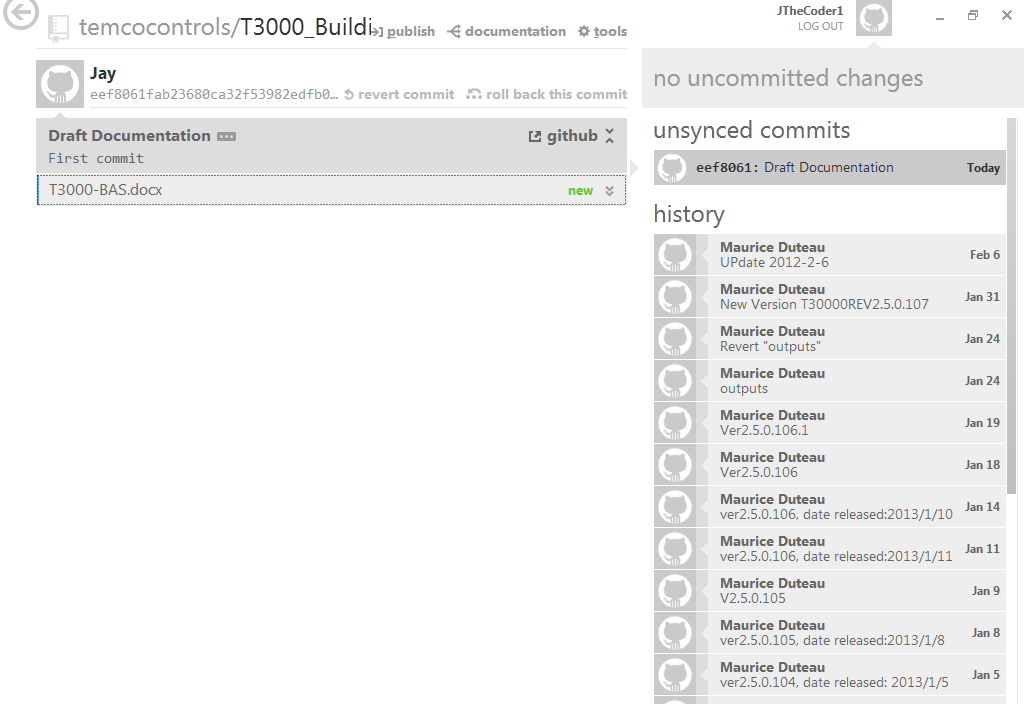
1. The repository starts getting cloned on local PC. The application shows the progress of the same. Be patient. The repository, depending on its size, takes few minutes to download.



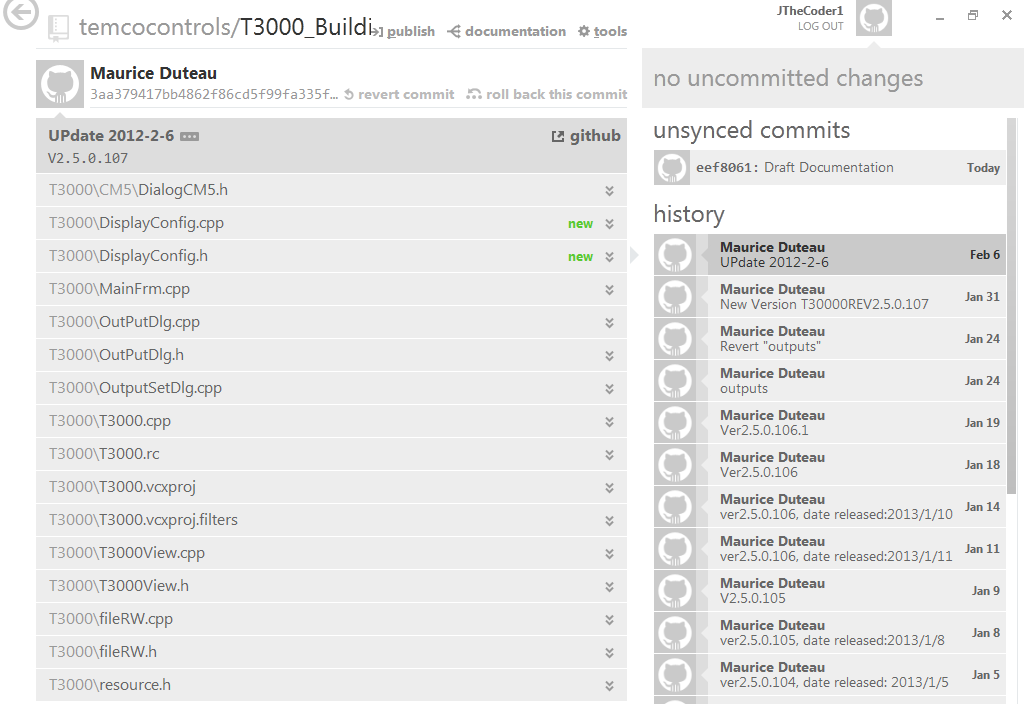
1. Once cloning is over, the repositories look like this.



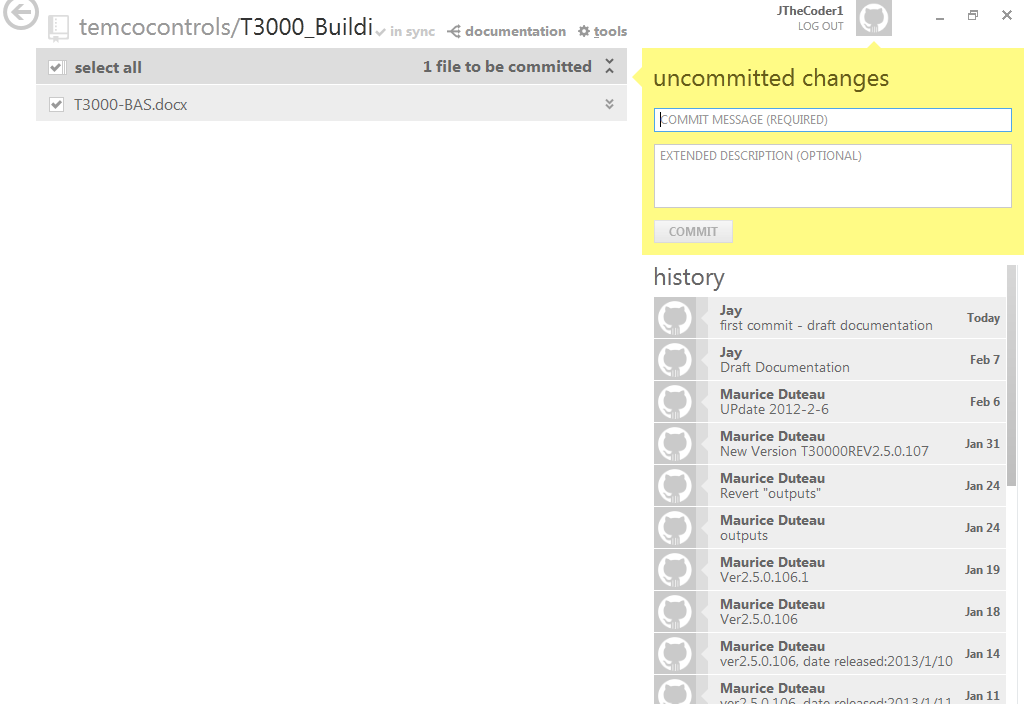
1. If you click on the small blue arrow on the repository, you would be taken to the following screen.



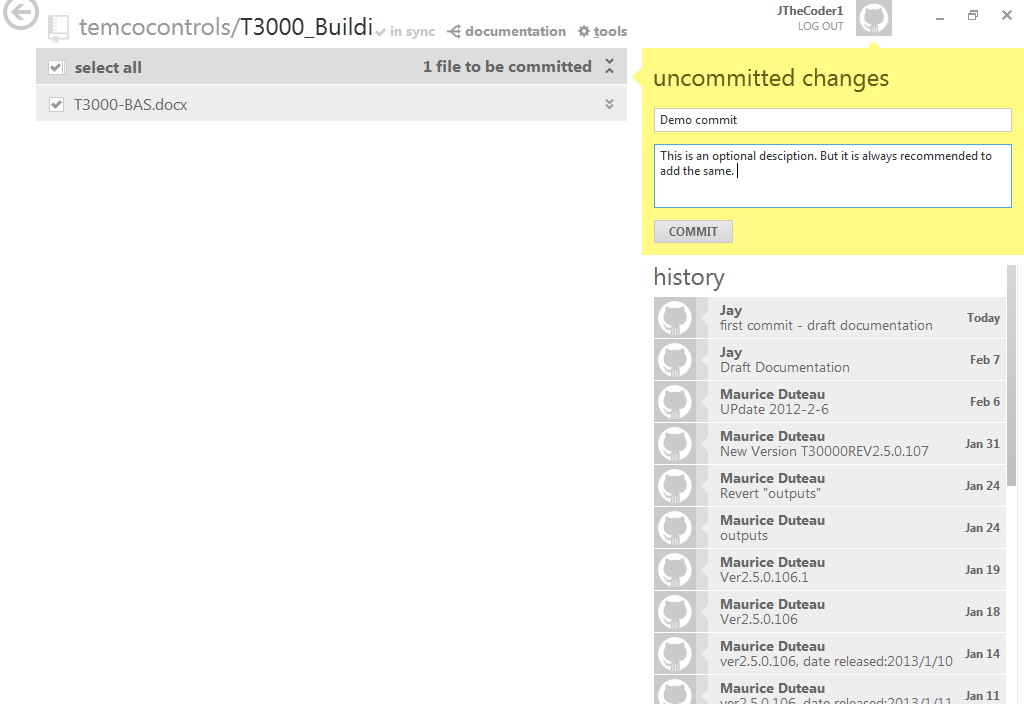
1. If you click on any of the links on history pane on right side, you get all the files that are there in the repository for that version.



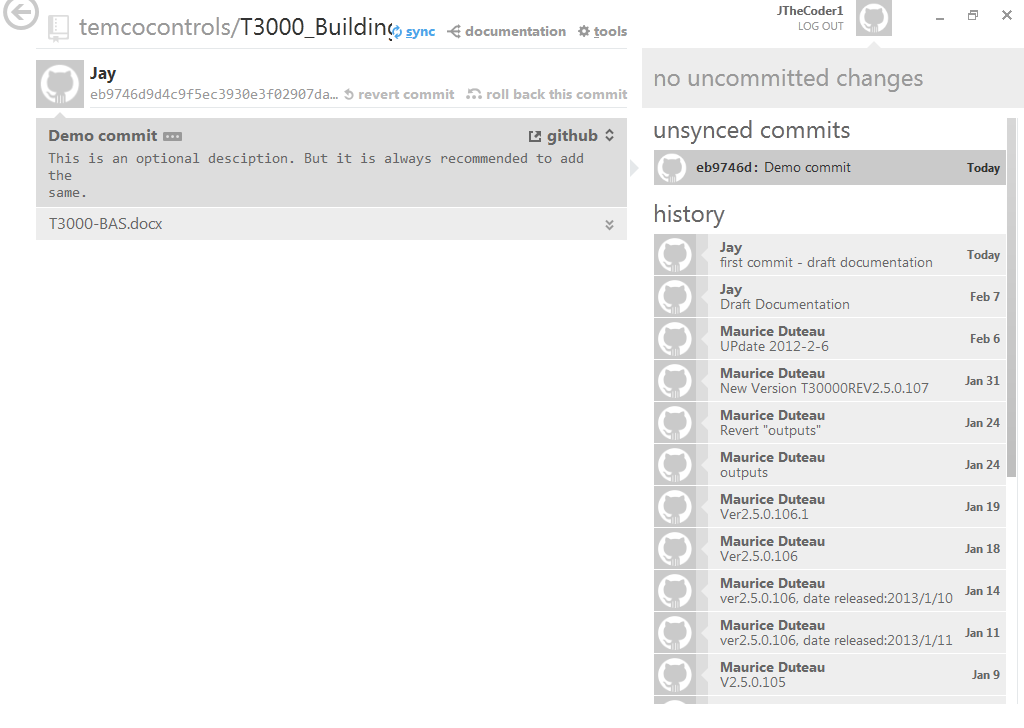
1. Once the file is modified in the repository, it appears on the screen as follows. I have the current file, which I am typing write now as the one which is to be committed to the repository. The same is shown on the screen.



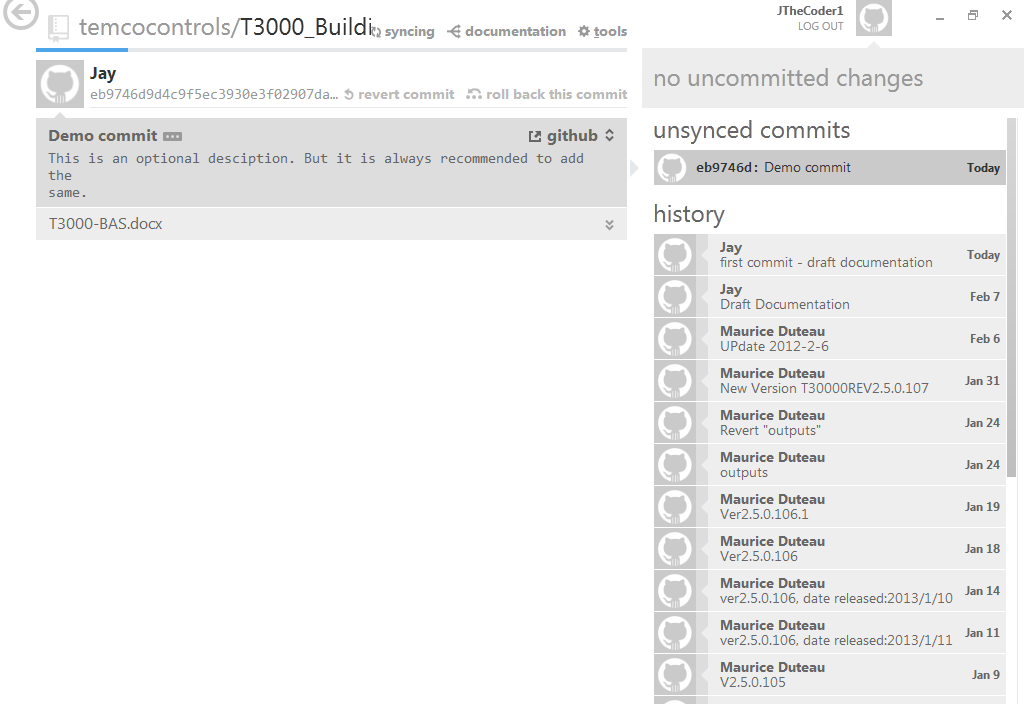
1. You can commit the change in the repository by adding a commit message and optionally an extended description. And then click on the COMMIT button given in the yellow area on right.



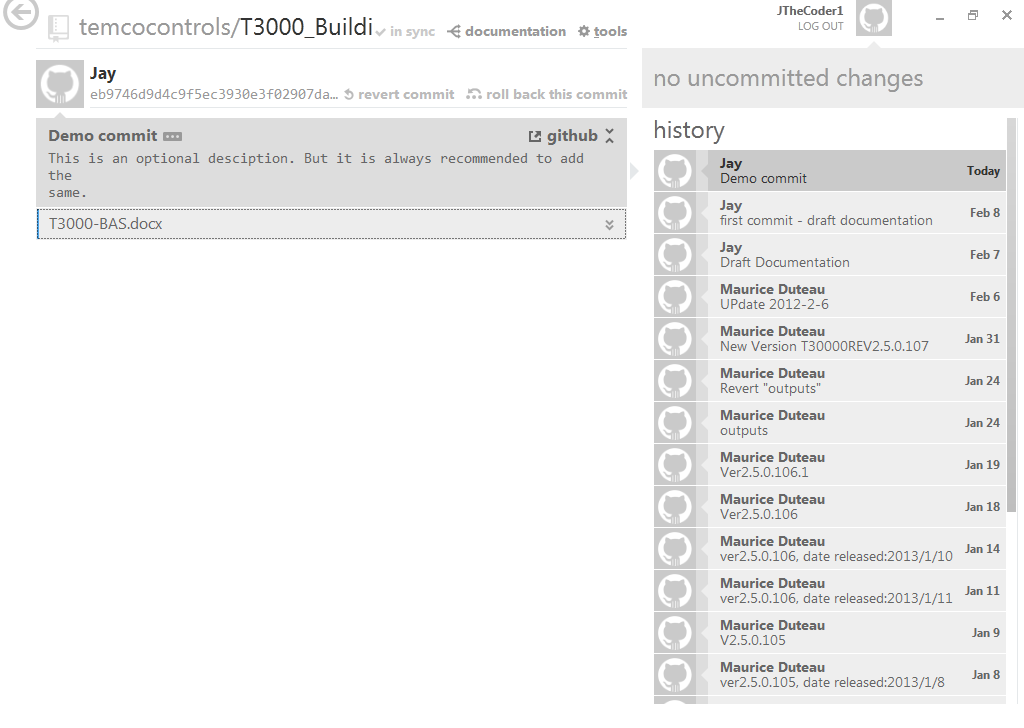
1. Once the COMMIT button is clicked, changes are committed to the repository. But they are not in sync with the repository on the github server. To push the changes to the github server, click on sync button as shown in blue in the following figure.



1. Following figure shows sync in progress.



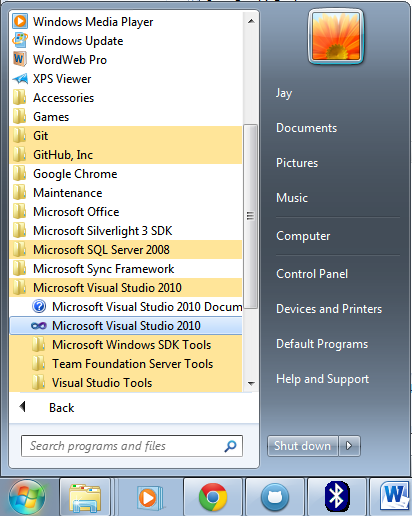
1. Once in sync, the same is shown as follows.



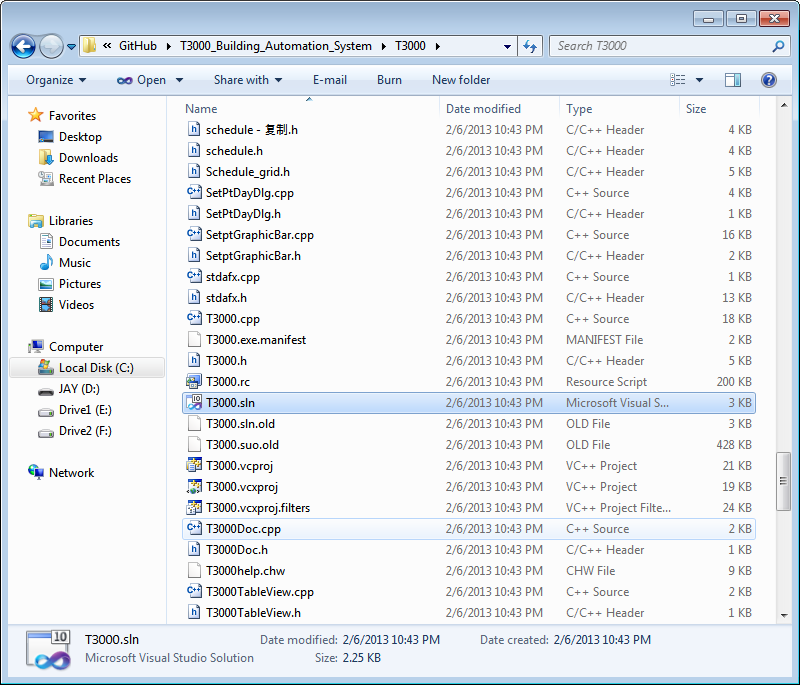
1. You can work with github and repository in this manner.

# Compilation on Microsoft Visual Studio 2010

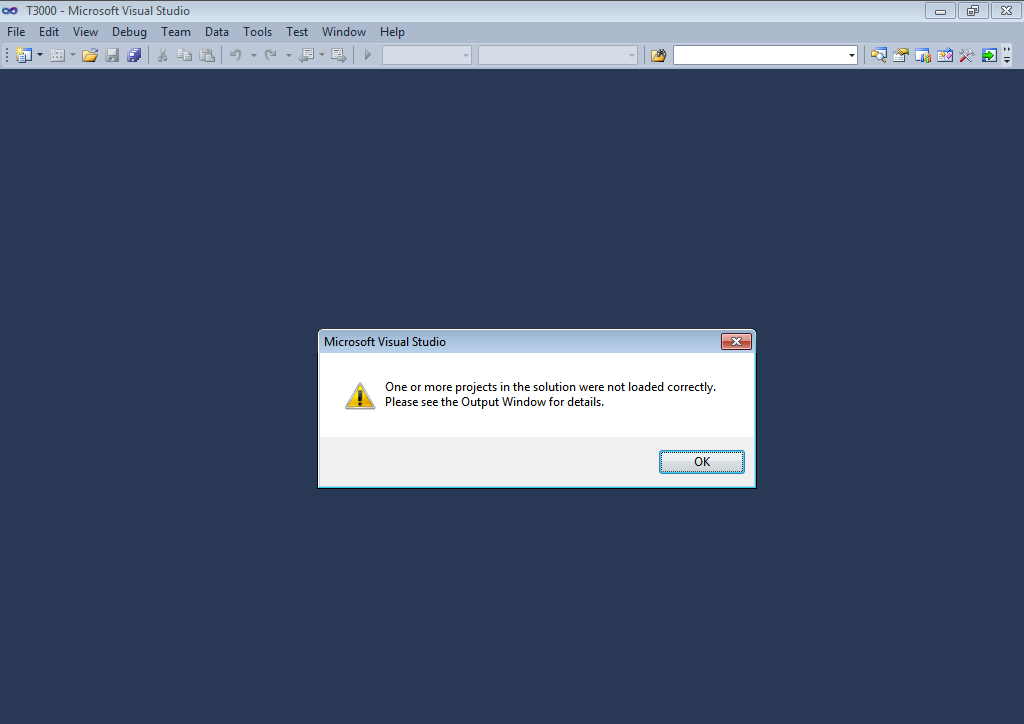
1. Install MS Visual Studio 2010.
2. Start MS Visual Studio from the main start menu.



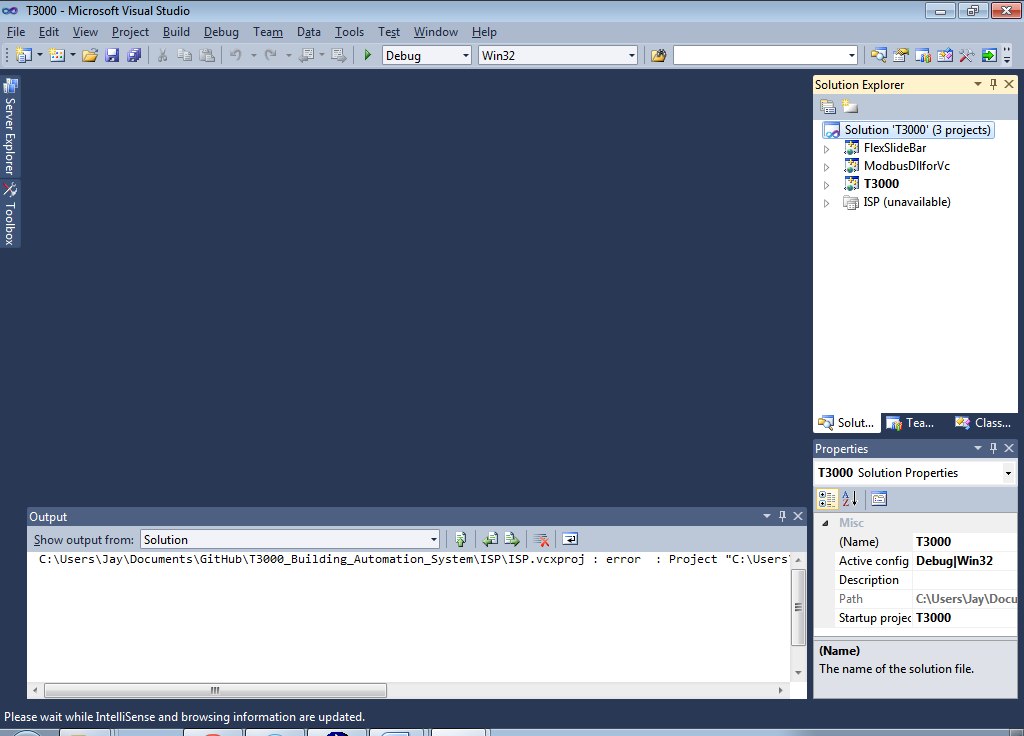
1. Go to the repository directory. Go to T3000 folder. Click on T3000.sln.



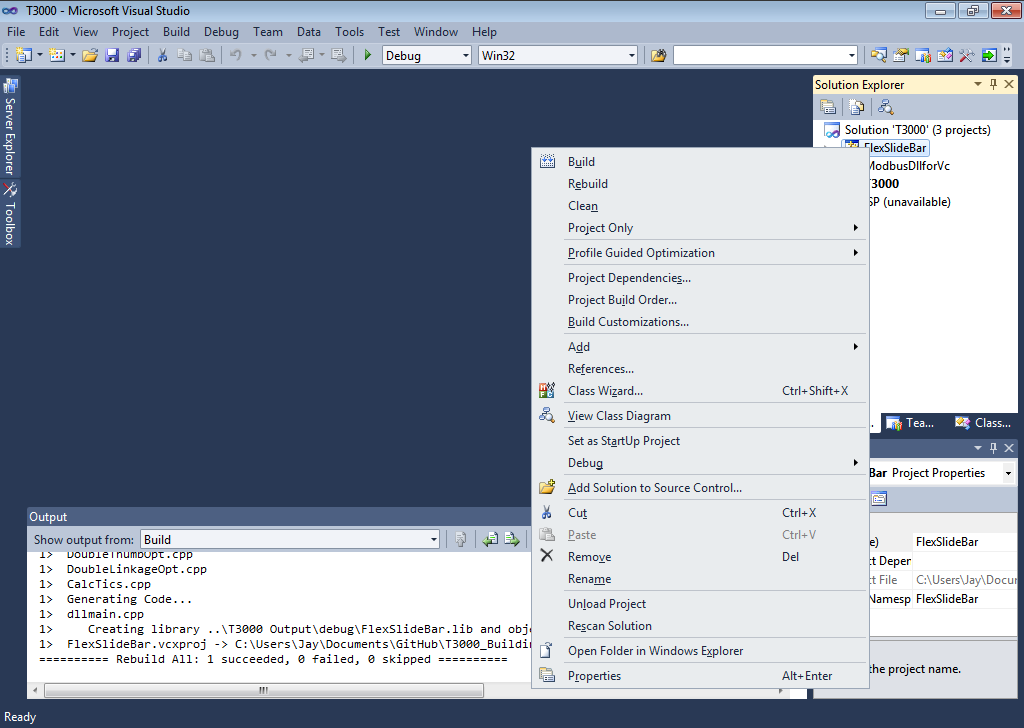
1. An error window appears as follows. [Right now, not sure why is this error coming? Anyway, capturing it in this doc as of now. We will remove this once the issue is solved. ]



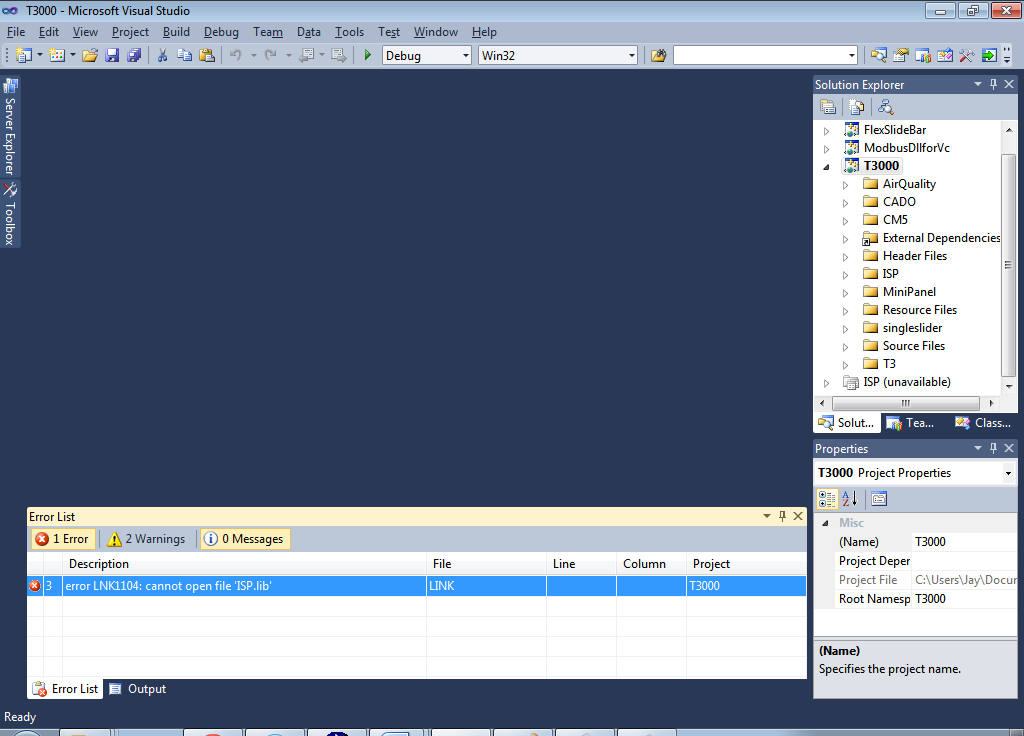
1. The following screen appears.



1. Right-click on FlexSlideBar project. And click Rebuild. This shall compile FlexSlideBar project.



1. Similarly compile ModbusDllforVc project by right-clicking it and clicking Rebuild.
2. Similarly compile T3000 project by right-clicking it and clicking Rebuild.
3. Right now, following linking error appears when we try to compile T3000. ISP.lib is not found.



1. Once, all projects are compiled, click CTRL + F5. This shall run the application and it will show application GUI as follows.

Not able to show this right now due to linking error.