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).

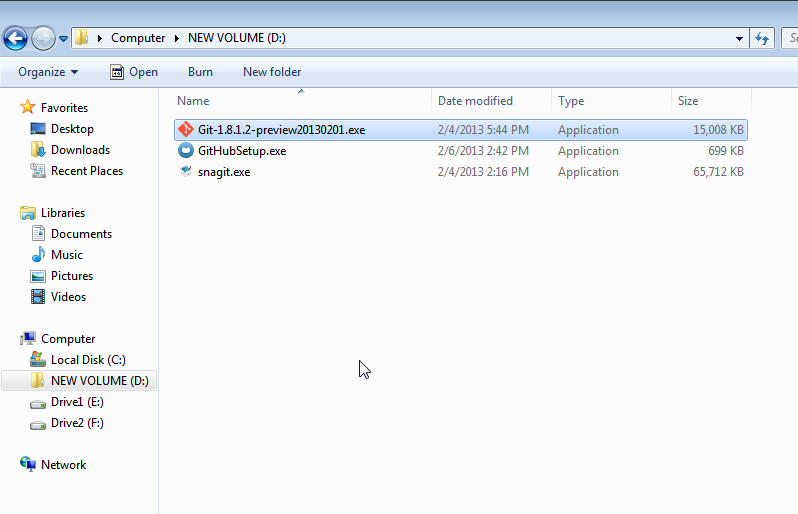
## Branching and Merging

## How to Commit in Git

# Installing Tools

Installation of git

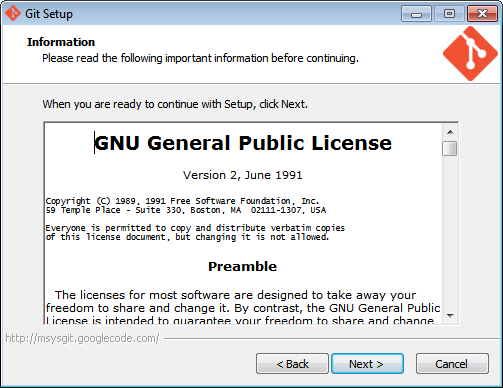
1. Download git installer.
2. Double click on the setup file.



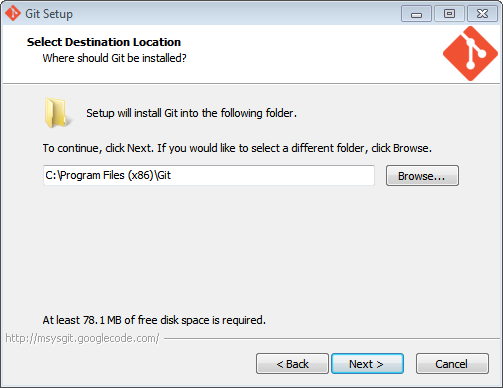
1. Following dialog box opens. Click Next to continue.



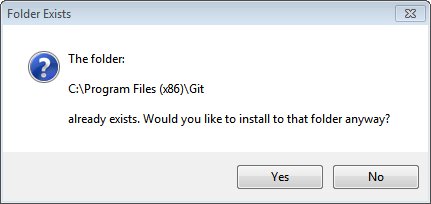
1. Read GNU General Public License. Click Next.



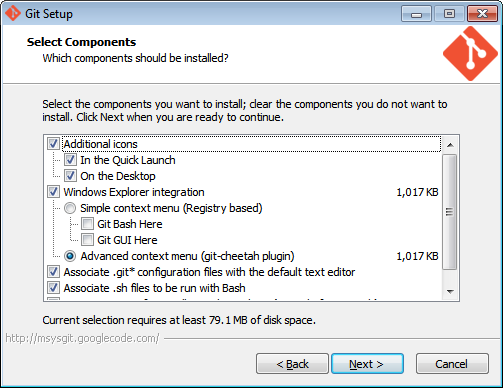
1. Select destination location. This is the directory where you want to install git. Normally, default directory can be kept as it is. If you wish, you can change the directory by clicking on Browse button. Here, we will install git in the default directory. (C:\Program Files (x86)\Git). Click Next.



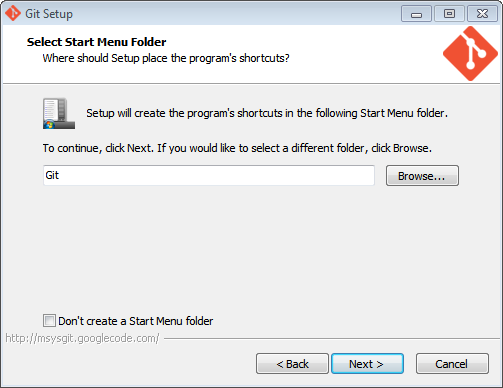
1. If following dialog box appears, click Yes.



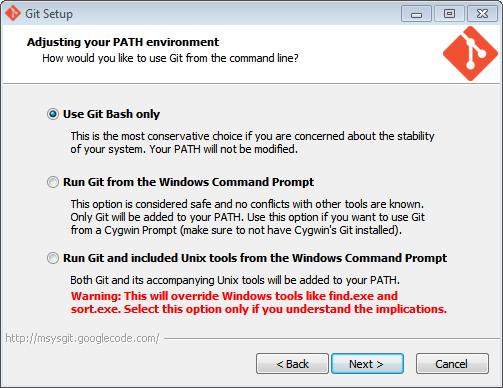
1. Following screen appears. Leave all the default options as is and click Next.



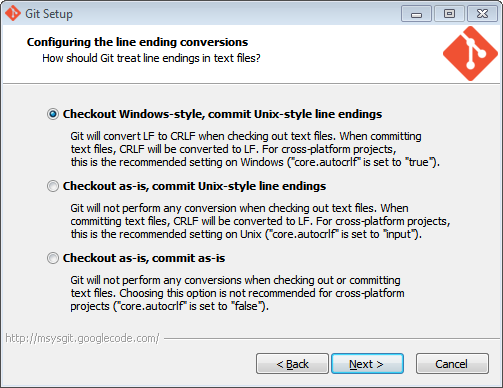
1. Following screen appears. Click Next.



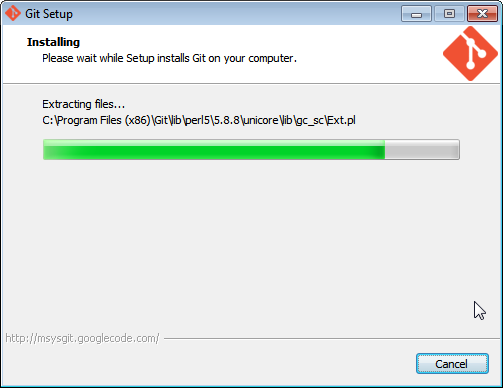
1. Following screen appears. Click Next to continue.



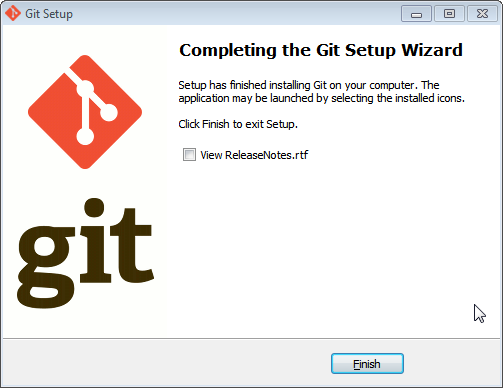
1. Following screen appears. Click Next to continue.



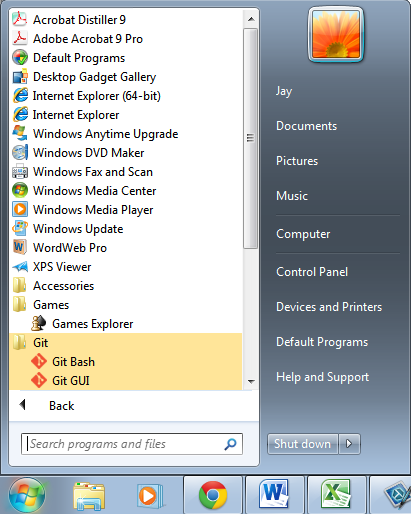
1. Setup will progress.



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

This is a command-line interface to work with Git and Git repositories.

1. Git GUI

This is a Graphical User Interface to work with Git and Git repositories.

1. Asd
2. Asd

# Introduction to GitHub