**How to install and use Git Extension on Window**

1. Introduction

Git Extensions is a toolkit aimed at making working with Git under Windows more intuitive (note that Git Extensions is also available on Linux and Macintosh OS X using Mono). The shell extension will integrate in Windows Explorer and presents a context menu on files and directories. There is also a Visual Studio plug-in to use Git from the Visual Studio IDE.

1. Download

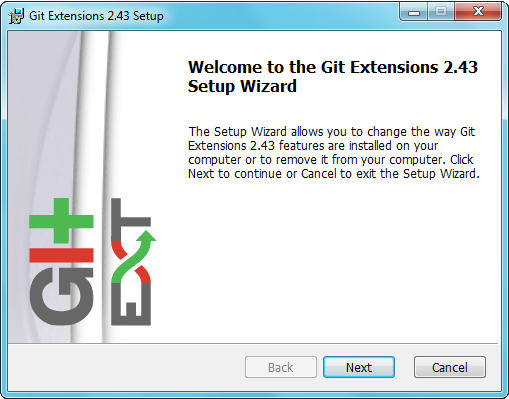
You can download Git Extension by visiting its download page below:

<https://github.com/gitextensions/gitextensions/releases/tag/v2.48.05>

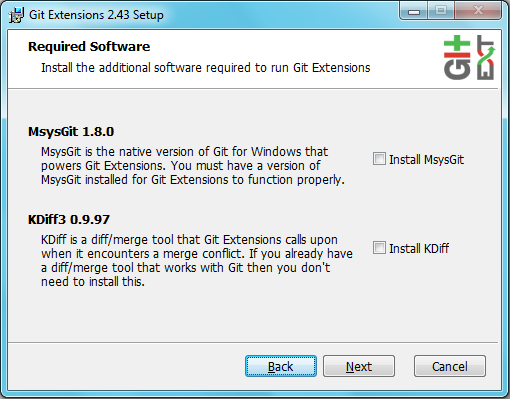
There are couple of installers for you to download. If you have not used Git Extension before, you can go ahead and download “[**GitExtensions-x.xx.xx-SetupComplete.msi**](https://github.com/gitextensions/gitextensions/releases/download/v2.48.05/GitExtensions-2.48.05-SetupComplete.msi)”. This installer will install Git for Windows 32bit, Kdiff3 32bit and Git Extensions on your computer.

1. Installation

After the file is downloaded, you can start the installation process by double clicks on the installation file.

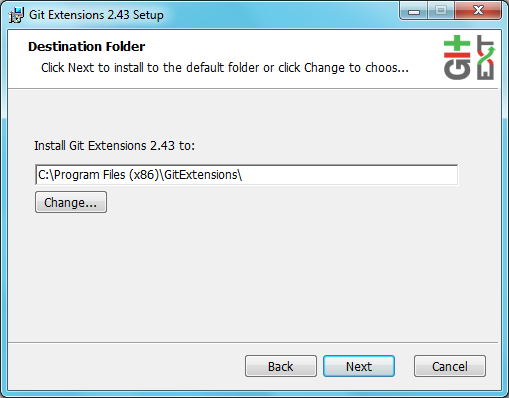


You can click “Next” to proceed to next step.

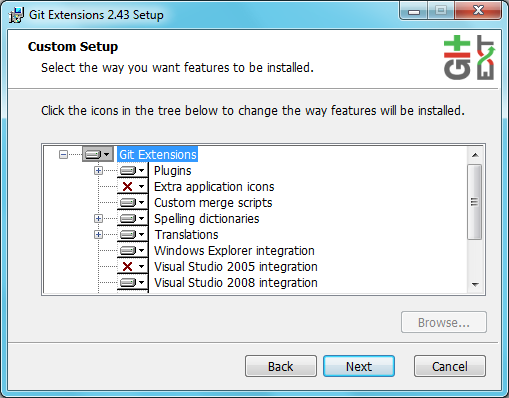


Make sure the options, MsysGit x.x.x and KDiff3 x.x.xx, are checked.

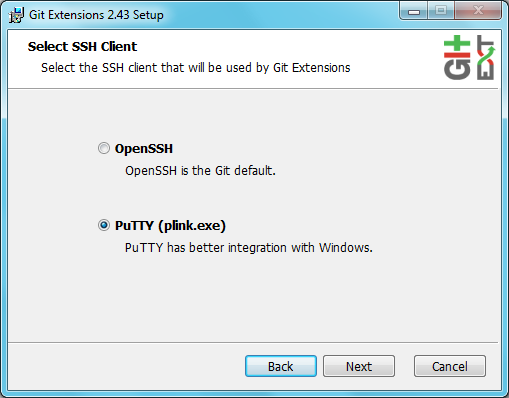
Note: The installation windows for MsysGit and KDiff will appear. You can follow the instructions on the windows to complete the process.

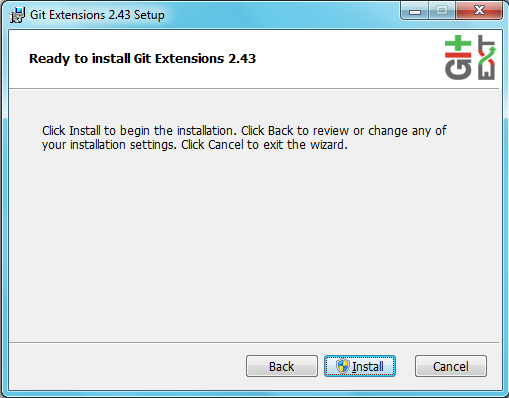


Please select a directory that you wish to install GE to.



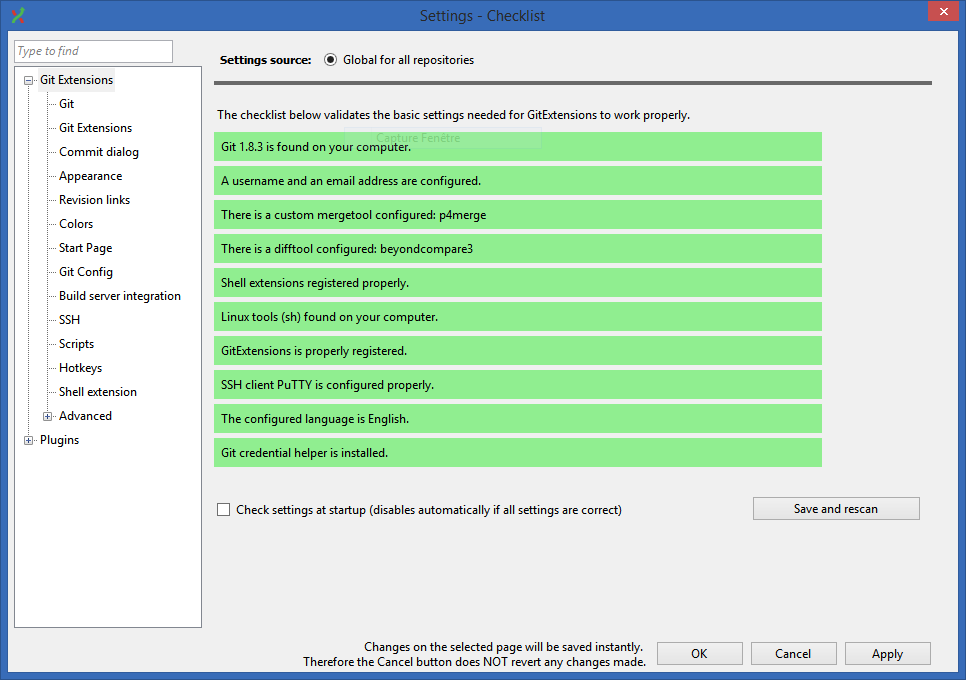
You can remove or install other features that you want. If you don’t know, then leave it as default.



Check OpenSSH option

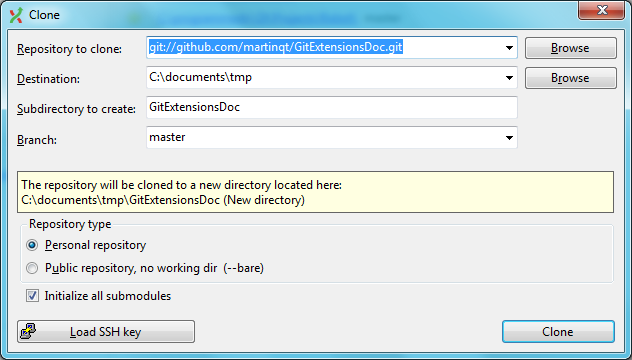
Click Install to start the installation.

When you open the program for the first time, the Setting window will appear. It asks you to fill out some information that is necessary for GE to run. You can click on “Repair” button to fix the problems. When all the problems are fixed, you can click Ok to enter the program and enjoy GE.



1. Clone/Create repository

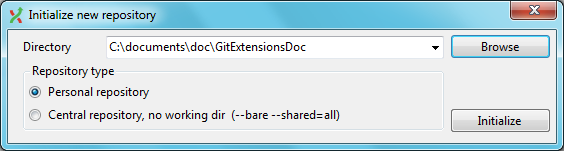
* You can clone an existing repository using this option. It displays the following dialog



The repository you want to clone could be on a network share or could be a repository that is accessed through an internet or intranet connection. Depending on the protocol (http or ssh) you might need to load a SSH key into PuTTY. You also need to specify where the cloned repository will be created and the initial branch that is checked out. If the cloned repository contains submodules, then these can be initialised using their default settings if required.

There are two different types of repositories you can create when making a clone. A personal repository contains the complete history and also contains a working copy of the source tree. A central repository is used as a public repository where developers push the changes they want to share with others to. A central repository contains the complete history but does not have a working directory like personal repositories.

* When you do not want to work on an existing project, you can create your own repository using this option.



Select a directory where the repository is to be created. You can choose to create a Personal repository or a Central repository.

A personal repository looks the same as a normal working directory but has a directory named .git at the root level containing the version history. This is the most common repository.

Central repositories only contain the version history. Because a central repository has no working directory you cannot checkout a revision in a central repository. It is also impossible to merge or pull changes in a central repository. This repository type can be used as a public repository where developers can push changes to or pull changes from.

1. Pull
2. Commit
3. Push
4. Merge branch