

git

Git and GitHub Training

This 90 Page PDF book on Git and GitHub will give you hands-on demo on how to use Git and GitHub practically for any project.

Last Updated on : 11/11/2016 9:40:32 PM

Sample copy

Created by Abhishek Luv

<http://www.abhishekluv.in/>

What is Version Control?

What is version control, and why should you care? Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

History of Git

- birth in 2005 by Linus Torvalds
- For Linux kernel maintenance

Installing Git and Git Extension

- msysgit.exe : Installs Git Software
- git extensions : GUI tool for using Git on PC

Welcome to the Git Setup Wizard

This will install Git version 2.7.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.



Next >

Cancel

Welcome to the Git Extensions 2.48.05 Setup Wizard

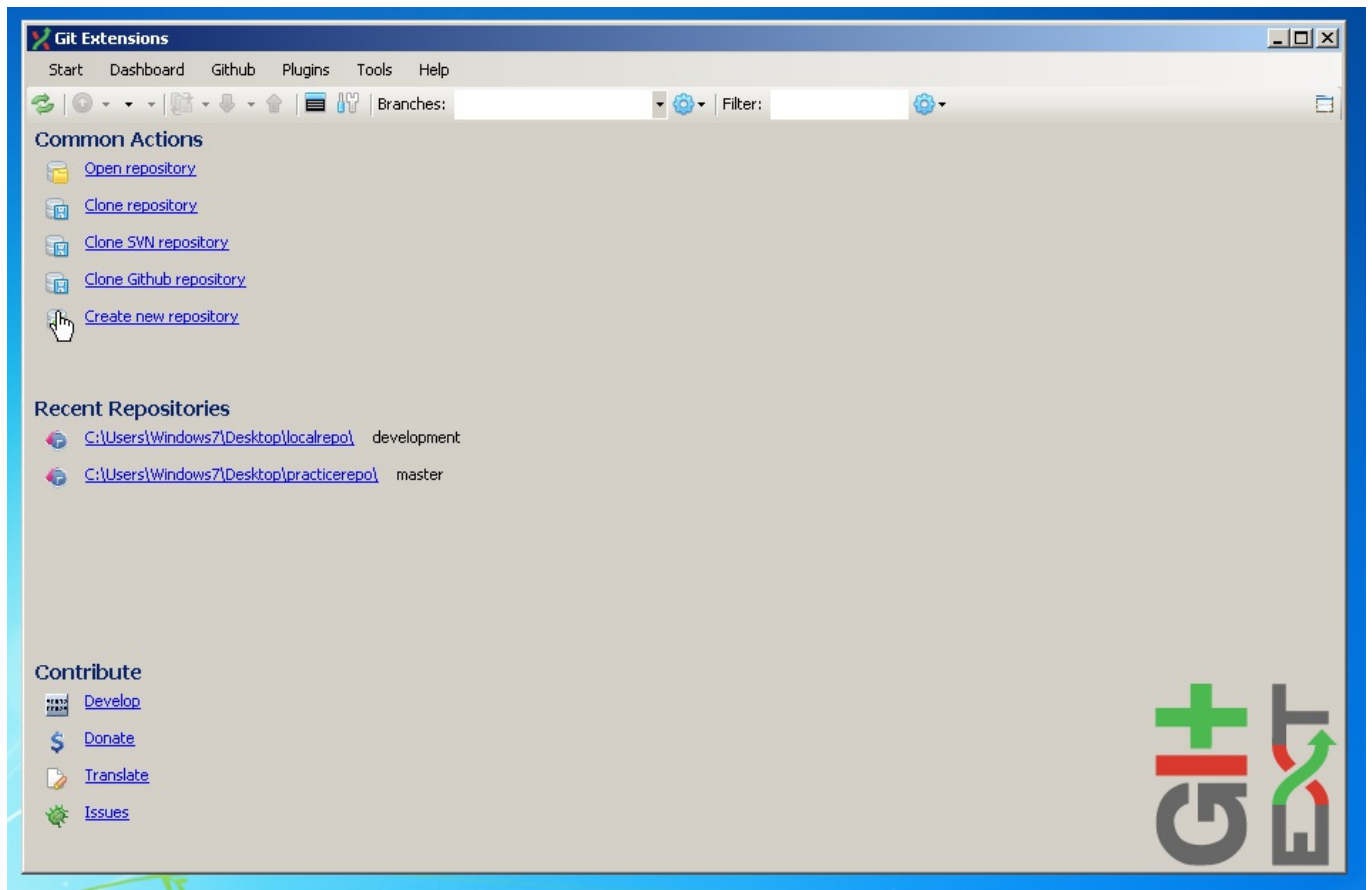
The Setup Wizard allows you to change the way Git Extensions 2.48.05 features are installed on your computer or to remove it from your computer. Click Next to continue or Cancel to exit the Setup Wizard.



Back

Next

Cancel



What is a repository?

A repository is simply a database containing all the information needed to retain and manage the revisions and history of a project.

Within a repository, Git maintains two primary data structures, the object store and the index.

The object store is designed to be efficiently copied during a clone operation as part of the mechanism that supports a fully DVCS.

Index is transitory information, is private to a repository and can be created or modified on demand as needed.

Content-Addressable Names

The Git object store is organized and implemented as a content-addressable storage system. Each object in the object store has a unique name produced by applying SHA1 to the contents of the object, yielding an SHA1 hash value. Any tiny change to a file causes the SHA1 hash to change, causing the new version of the file to be indexed separately.

Git is a content-addressable filesystem.

What does that mean?

It means that at the core of Git is a simple key-value data store.

You can insert any kind of content into it, and it will give you back a key that you can use to retrieve the content again at any time.

SHA1 values are 160-bit value that are usually represented as a 40 digit hexadecimal number.

For example: `937082042f48a5cbc7777634509310fff059bc19`

Understanding what is an Index in Git

The index is a temporary and dynamic binary file that describes the

directory structure of the entire repository.

Index captures a version of the project's overall structure at some moment in time. As a developer, we execute Git commands to stage changes in the index.

Changes usually add, delete or edit some files or set of files. The index records and retains those changes, keeping them safe until you are ready to commit them. You can also remove or replace changes in the index.

Git's Index doesn't contain any file content: it simply tracks what you want to commit when you run `git commit`.

Git checks the index rather than your working directory to discover what to commit.

You can query the state of the index at any time with the command `git status`

File Management & Classification in Git

Remote Repository(master) <-> Index(Staging happens here) <->
Local Repository (local)

- Tracked : Any file that's already there in the repository or any

file that is staged

- Ignored : file to be ignored by git object store and git index
- Untracked : An untracked file is any file not found in either of the previous two categories

Git Internals : The .GIT Directory

.git Directory

This is the magic directory where all the git “stuff” is stored.

Git Object Store

You should see a bunch of directories with 2 letter names. The directory names are the first two letters of the sha1 hash of the object stored in git.

It contains your original data files and all the log messages, author information, dates, and other information required to rebuild any revision or branch of the project.

There are four types of Object stores:

- Blobs: A blob holds a file’s data but does not contain any metadata about the file or even its name.
- Trees: Its records blob identifiers, path names and a bit of metadata for all the files in one directory. It can also recursively reference other subtrees.

- **Commits:** A commits object holds metadata for each change introduced into the repository, including the author, committer, commit data, and log message.
- **Tags:** A tag object assigns an arbitrary human readable name to a specific object usually a commit.

Config file

This is a project-specific configuration file. Config entries in here will override the config entries in the `.gitconfig` file in your home directory, at least for this project.

Branches and Tags

You should recognize the files in the tags subdirectory. Each file corresponds to a tag you created with the `git tag` command earlier. Its content is just the hash of the commit tied to the tag.

The heads directory is similar, but is used for branches rather than tags. We only have one branch at the moment, so all you will see is master in this directory.

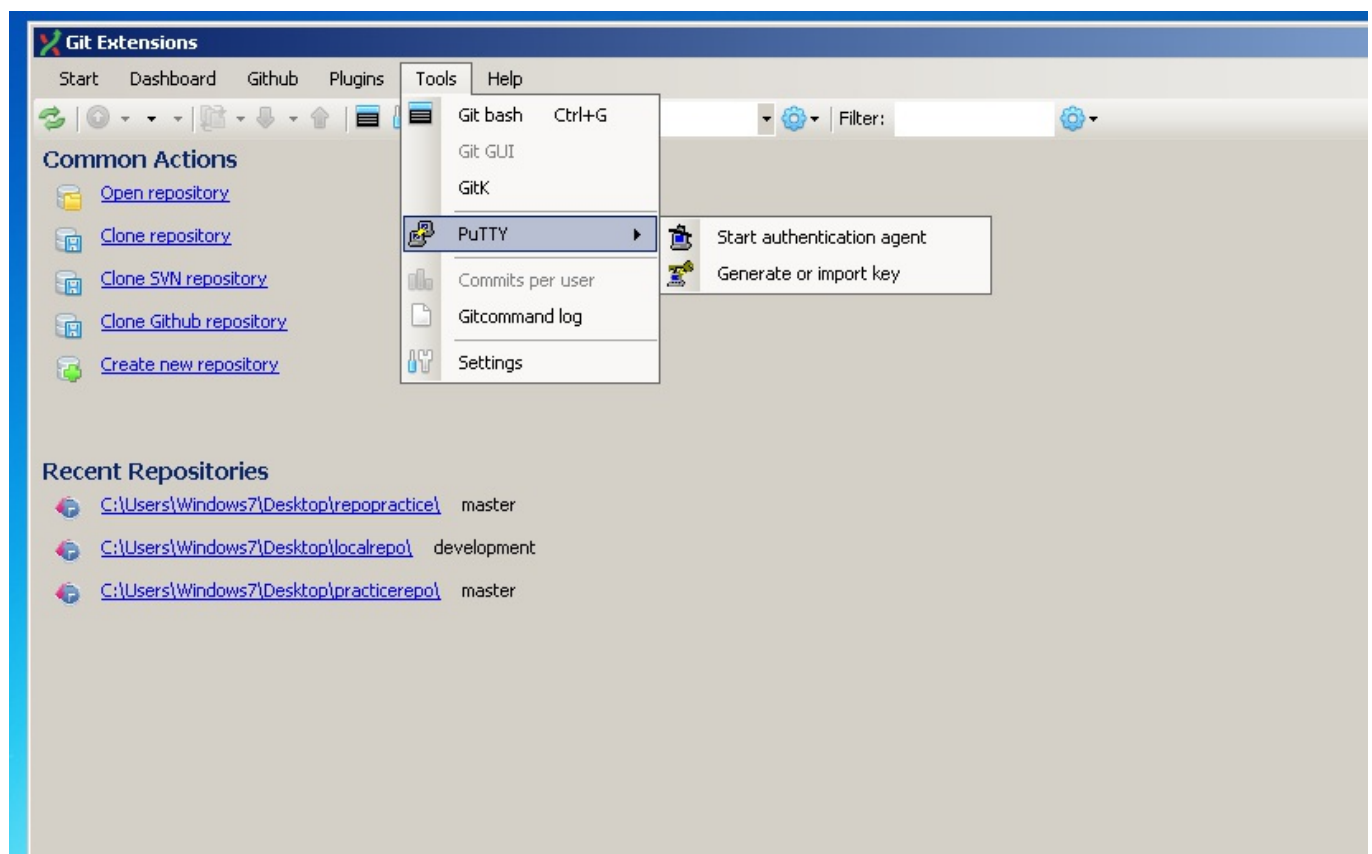
The HEAD File

The HEAD file contains a reference to the current branch. It should be a reference to master at this point.

Generating SSH Key for GitHub using Git Extensions

Using GitHub via HTTPS with Git Extensions can become pretty boring. SSH to the rescue. With SSH you won't have to enter the [GitHub.com](https://github.com) password again and again for every push you make.

Click on Generate or import key



Click on Generate button

