The Beginner's Guide for GIT Extensions: How to use GIT to clone repository from GitHub and make changes

Introduction to Git Extensions

GIT Extensions is a distributed version control system enabling user to robustly manage collection of source files and the changes made in them. The changes made are shown in History of changes. Users can make changes by accessing a Central repository called remote repository and committing the changes to it. It implements classic GIT by using GUI (Graphical user interface), basically driven by a set of dedicated commands, hence maintains the version control system intuitively. So, let us go through a glimpse of functionalities provided by GIT Extensions so that our version control system can be maintained.

Managing repository

There are many options to manage repository through GIT Extensions. It includes viewing the committed logs and changes made in comparison to previous commit, cloning a repository, traversing through the file directory and filtering the committed logs by using custom search input etc.

GIT Extensions can be downloaded from
https://github.com/gitextensions/gitextensions/releases/tag/v2.48.05

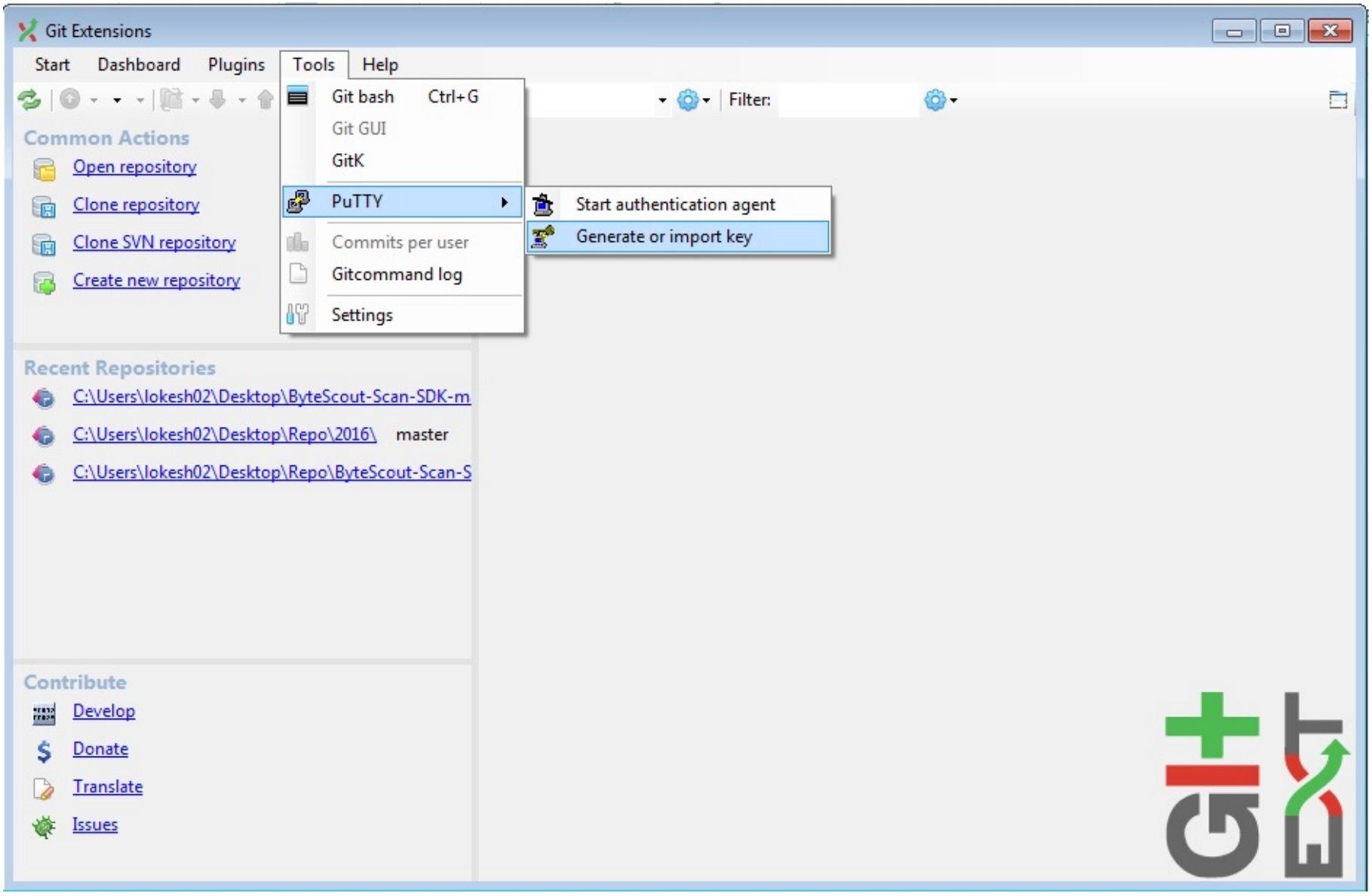
Generating SSH Keys as one time activity

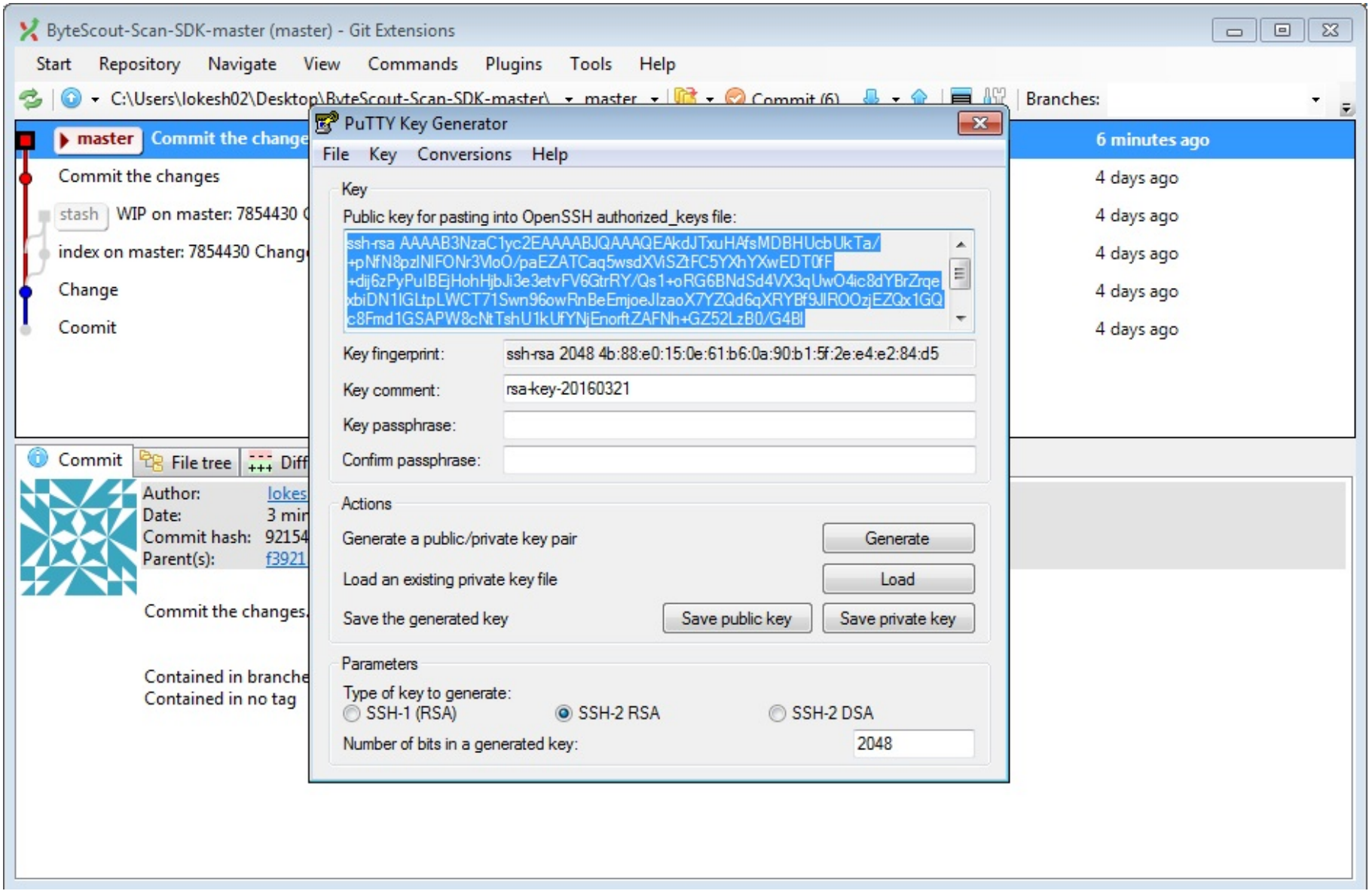
SSH Keys should be loaded as a one-time activity. SSH keys can be generated while setting up the GIT Extensions.

In order to use a safe development environment with SSH you need to get PuTTY installed as preferred SSH client. PuTTY can be downloaded from http://www.putty.org/

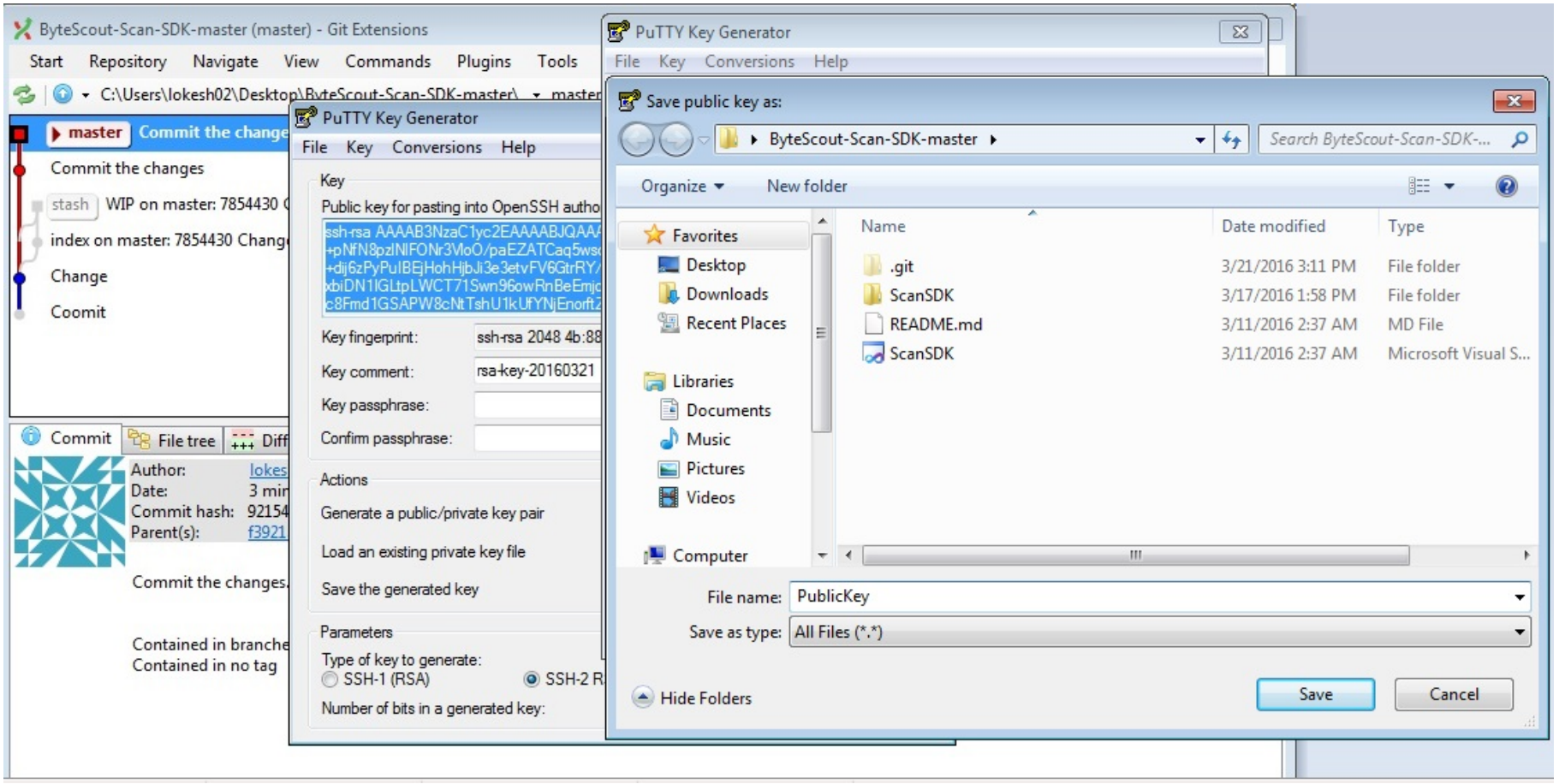
The first thing is to check that Git Extensions is properly configured to use PuTTY as well as all paths are given correctly.

In the Remotes Tab just choose **Generate or import key** to start the key generator.


Git Extensions
Start
Dashboard
Plugins
Tools
Help
Filter:
Common Actions
Open repository
Clone repository
Clone SVN repository
Create new repository
Git bash Ctrl+G
Git GUI
GitK
PuTTY
Commits per user
Gitcommand log
Settings
Start authentication agent
Generate or import key
Recent Repositories
C:\Users\lokesh02\Desktop\ByteScout-Scan-SDK-m
C:\Users\lokesh02\Desktop\Repo\2016\ master
C:\Users\lokesh02\Desktop\Repo\ByteScout-Scan-S
Contribute
Develop
Donate
Translate
Issues



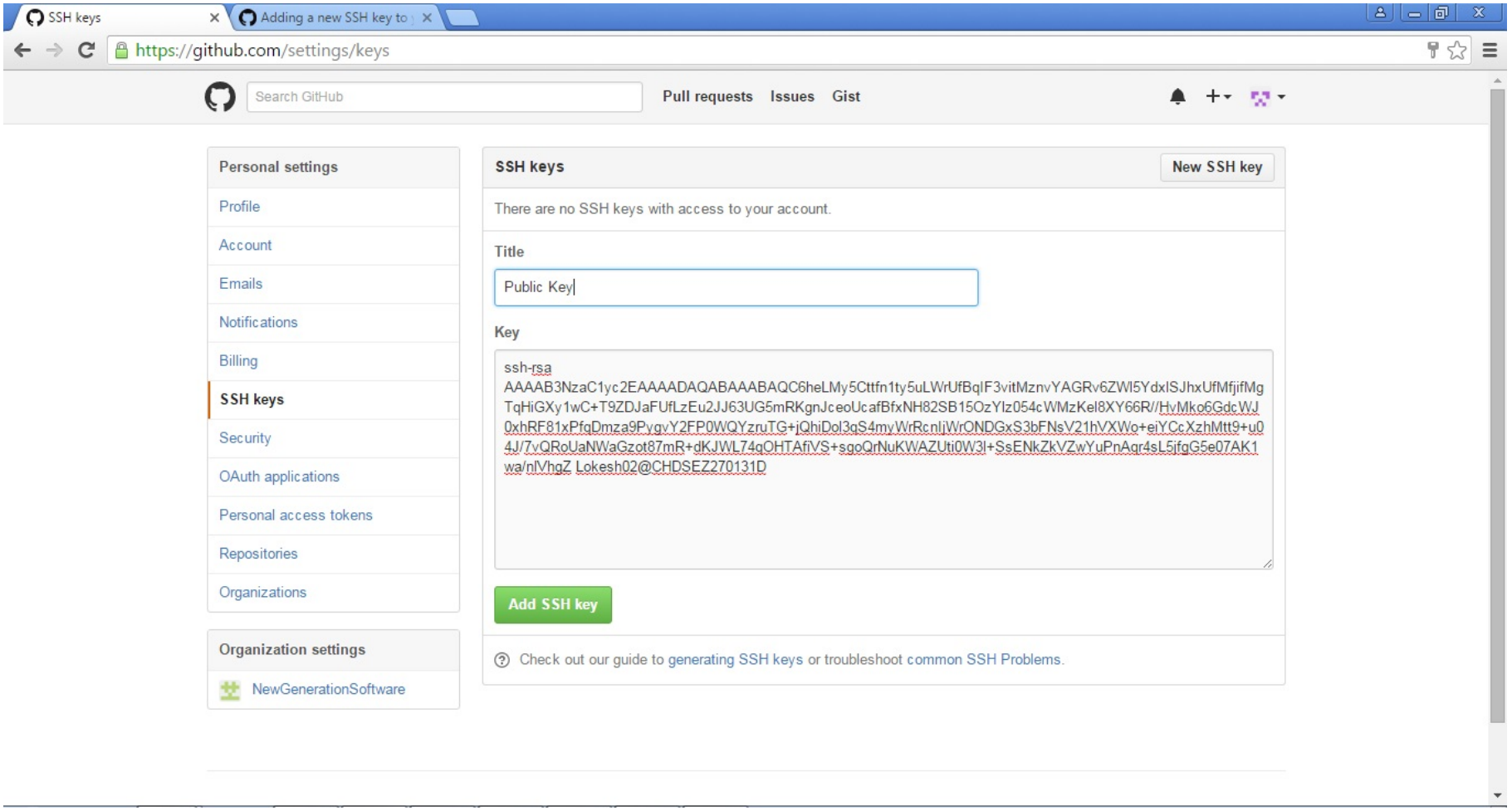
It will ask you to move around the mouse in order to generate more random keys. When the key gets generated, save the public and private key in a text file by clicking the Save Public key button.



Since now you are having a key pair, provide the public key to the Github account by copying the Key from the file which you just saved above at your desired location.

Then open your Github account and click on the profile image, followed by Account settings and going to the SSH Keys tab. Finally, paste the public key over there.

You can create a Github account at https://github.com/join?source=header



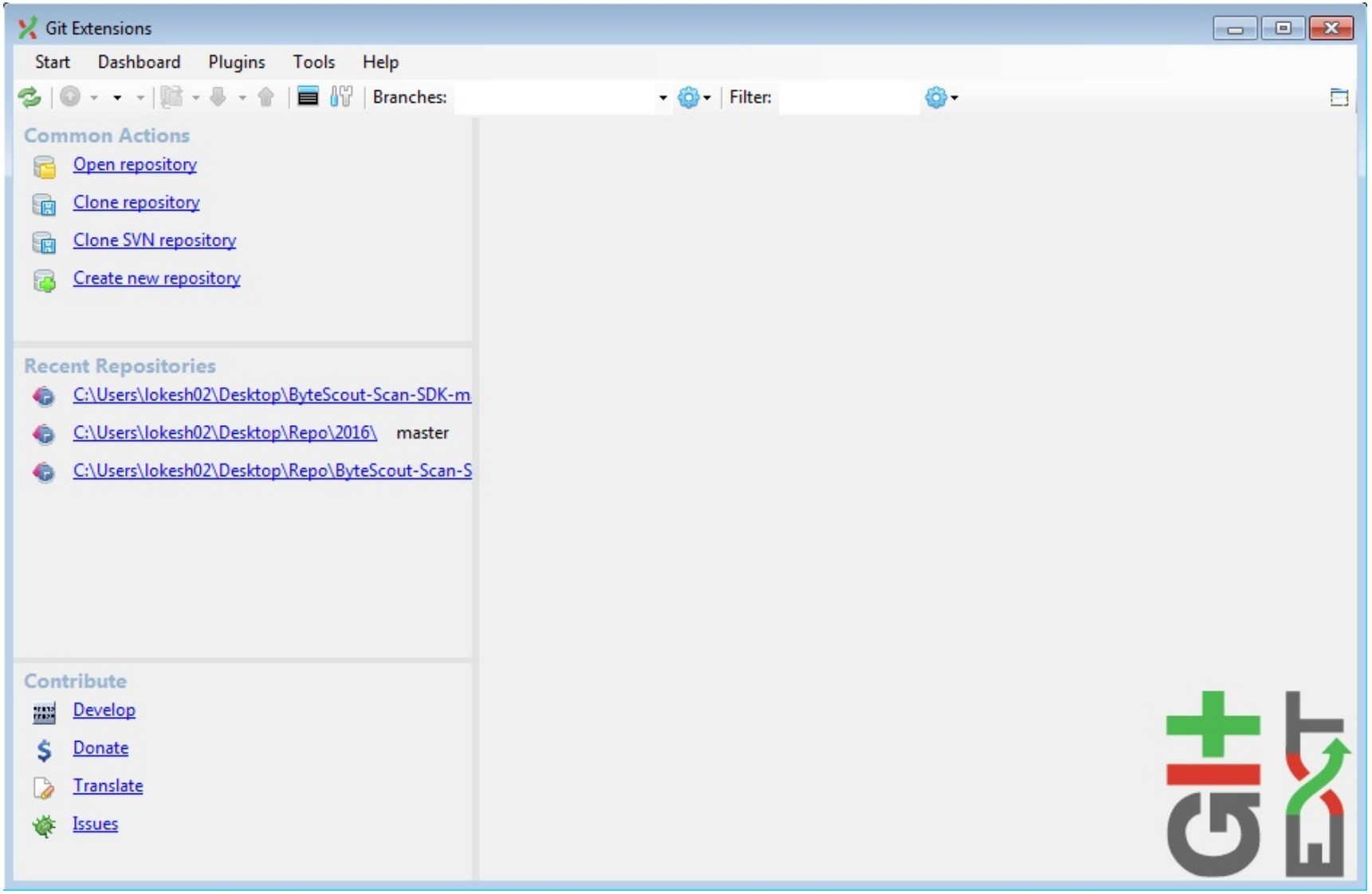
Now github will get to know which public key it has to use to decrypt. Now you also need to provide the private key to GitExtensions to encrypt. You will find a Load SSH key button in the clone dialog where you can load the private key in PuTTY authentication.

Note: This is a one-time activity and you don't need to repeat these steps again.

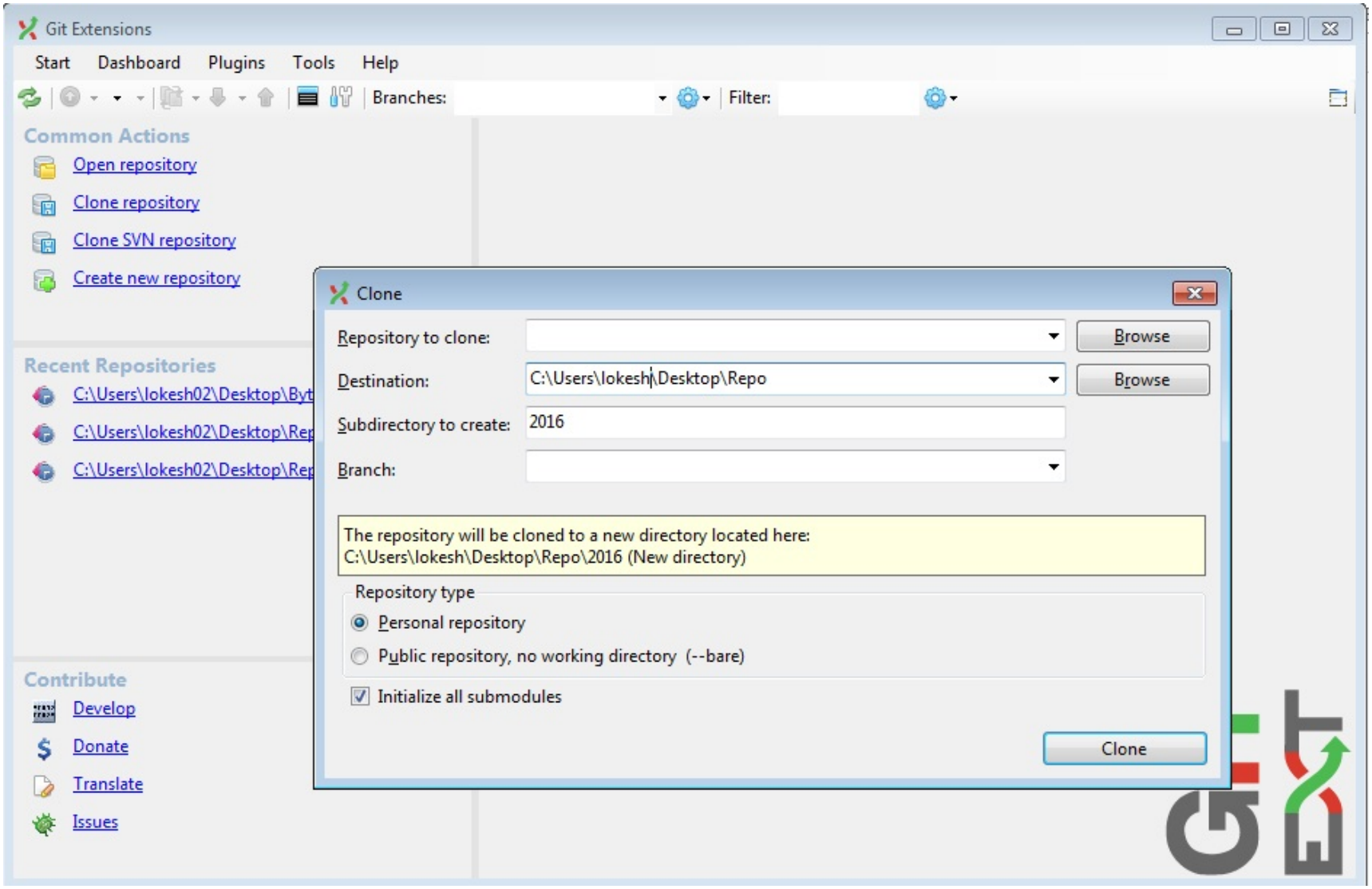
How to Clone a Repository?

Cloning a repository will create a local copy of the repository being cloned by this action.

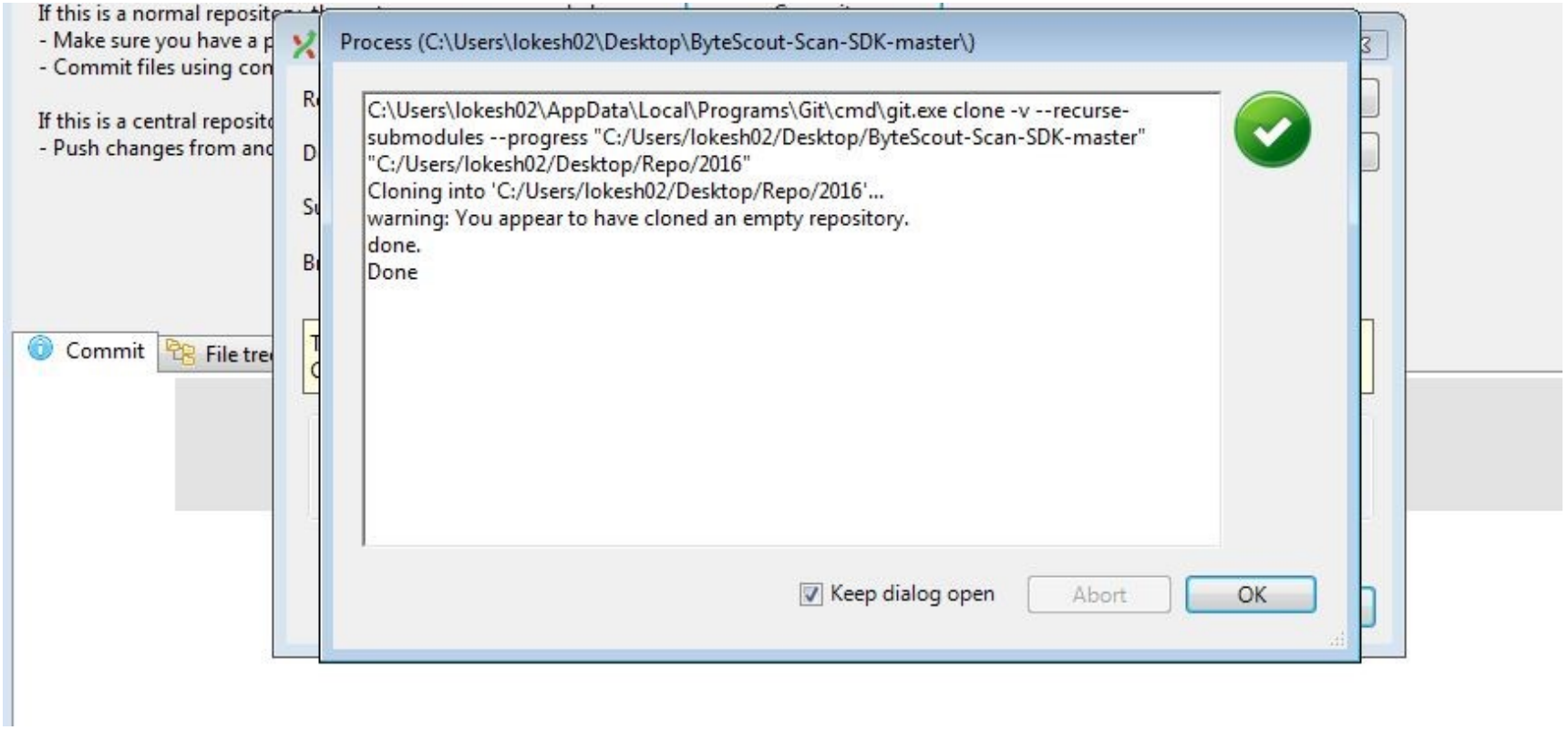
1. Click 'Clone repository' link from *Common Actions* section in the left to clone a repository.



2. Provide the necessary inputs as *Repository to clone (highlighted URL at the top tells which repository needs to be cloned), Destination (local hard drive location), Branch (automatically loaded when the local repository address is provided), and Repository type* and then click 'Clone'.



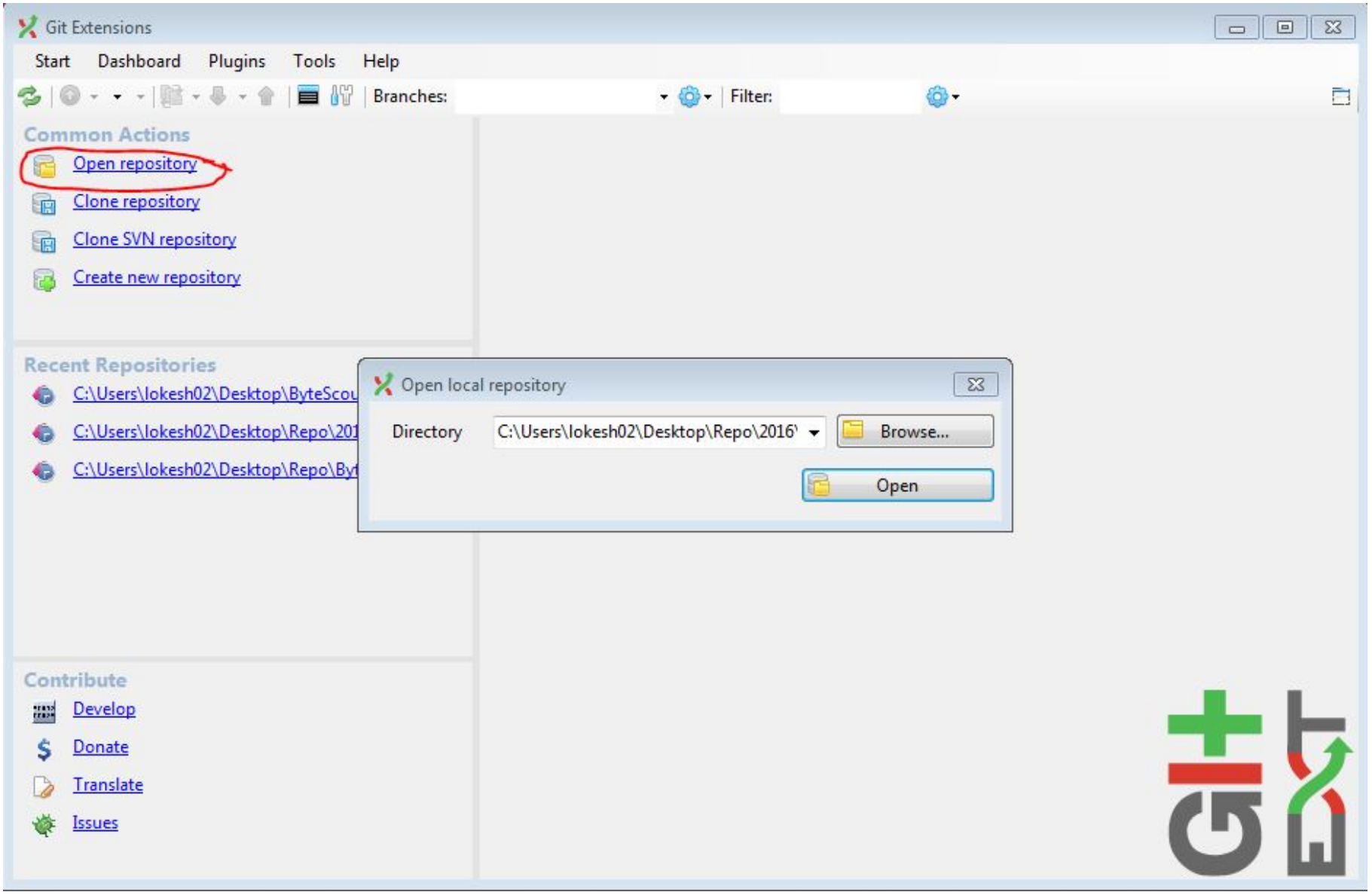
3. The process of cloning starts and the remote files are checked-out to the specified destination i.e. local directory. A green tick mark indicating completion of process will be displayed.



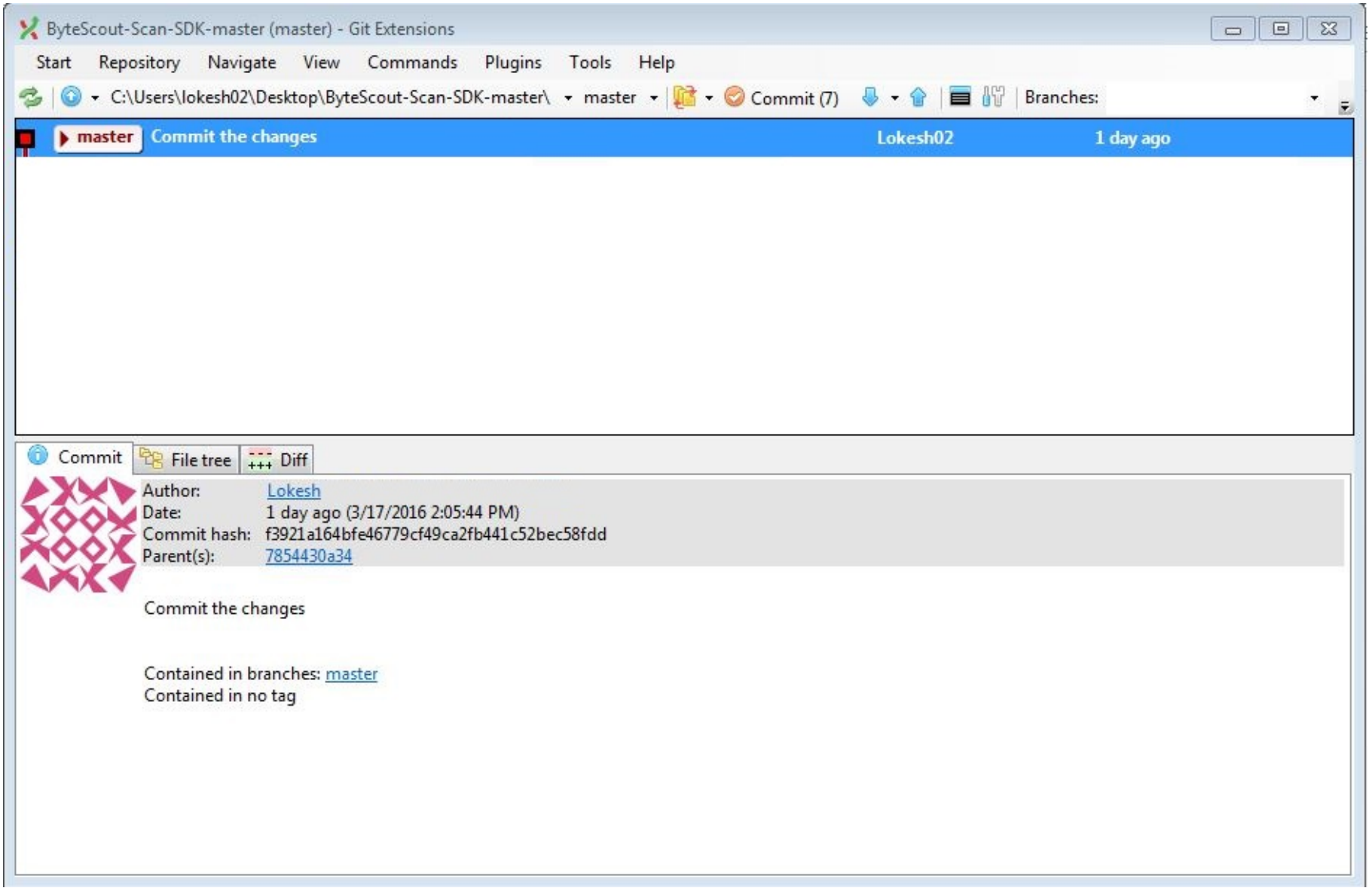
Note: A Red Cross mark with respective errors will be displayed too during occurrence of errors in the process, if any.

How to open a repository?

From the Common Actions section at the left click ‘Open Repository’ link and give the directory address for opening a repository. Then click ‘Open’ button as highlighted in the figure.

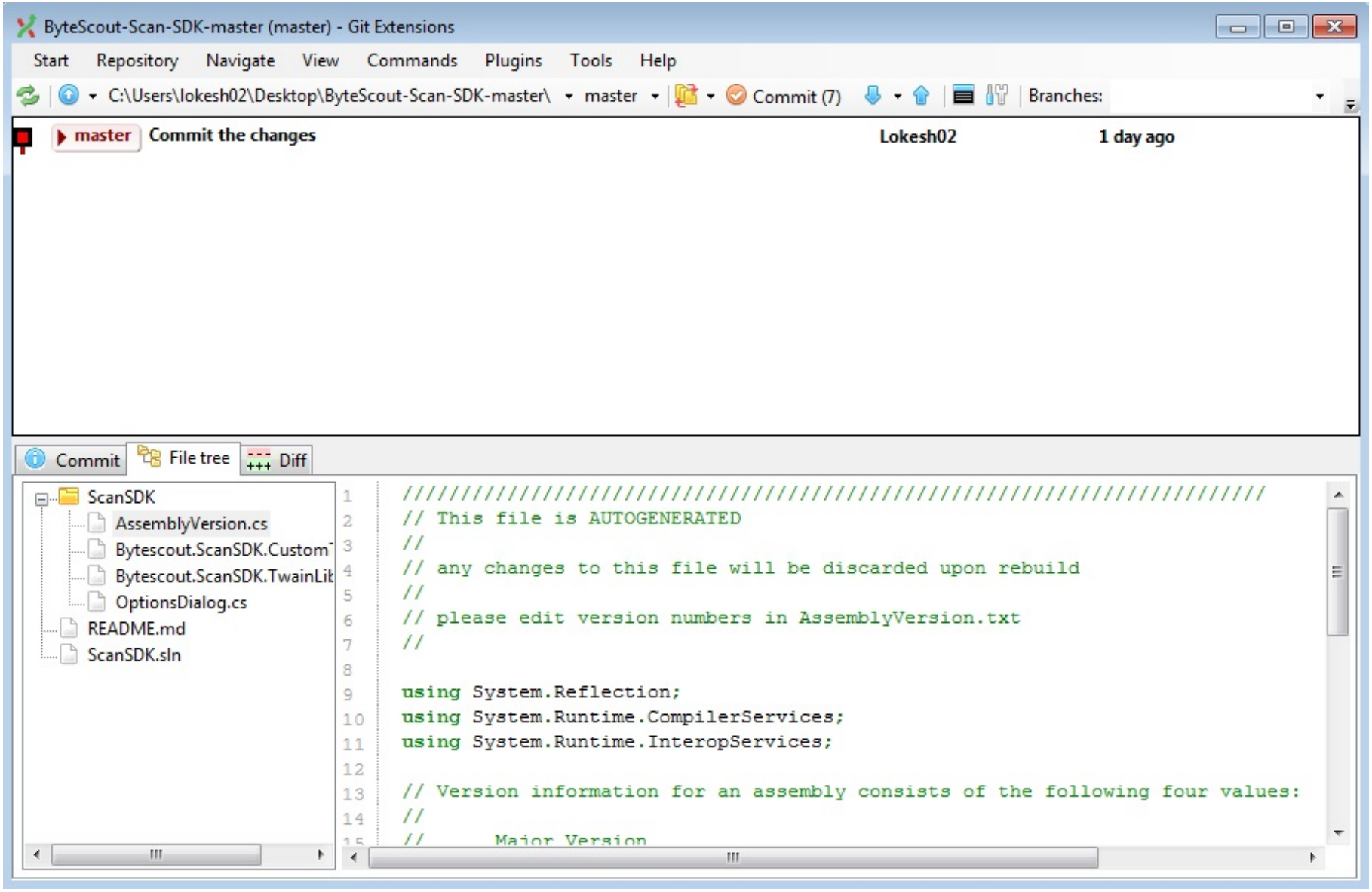


It opens the repository and all committed logs will be shown with abstract message associated with them, committed user and the time elapsed when the commit was done (Refer figure). Also, the details like Author, date are shown at the bottom in the commit section.



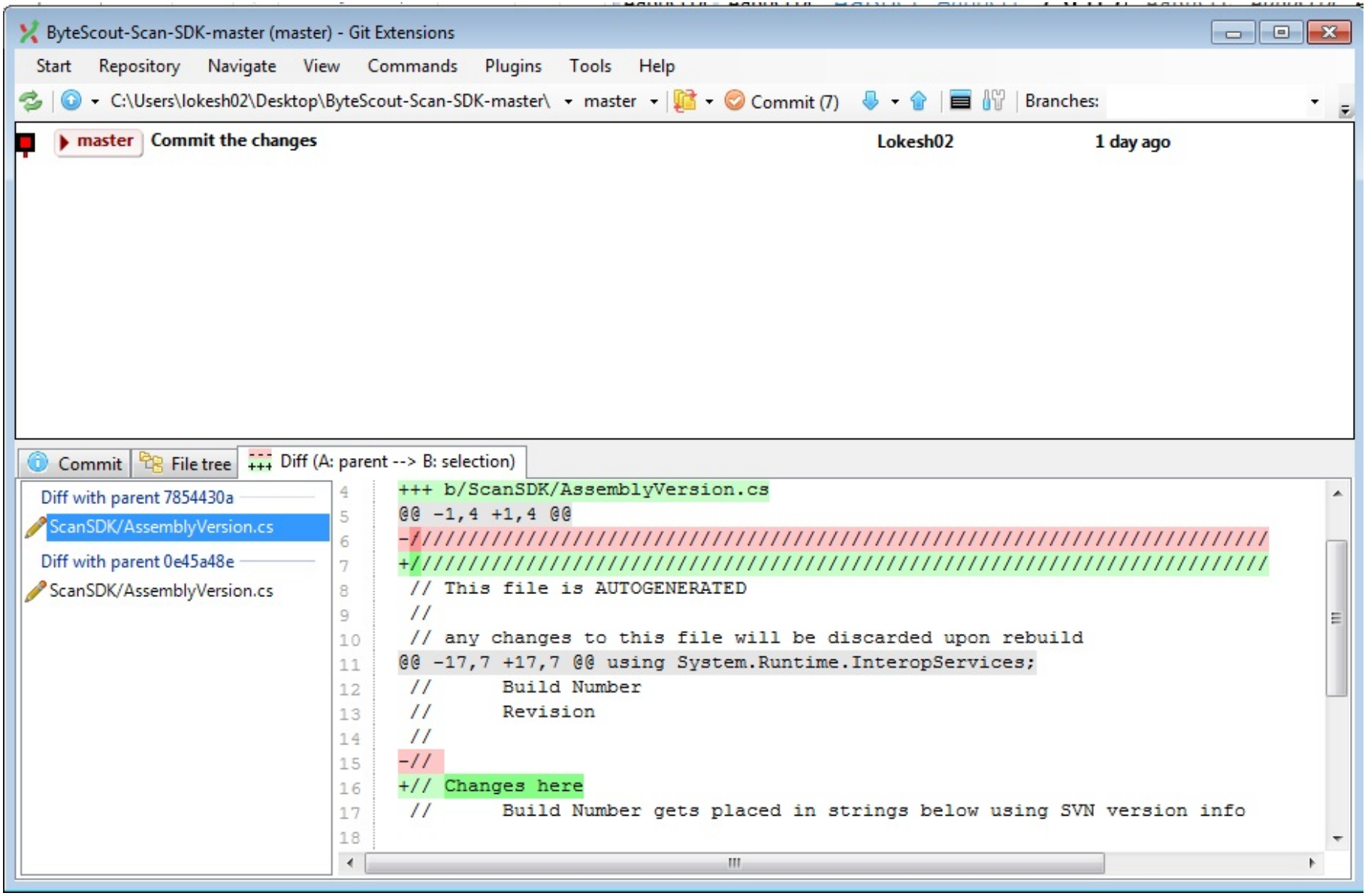
How to traverse into repository?

Initially click ‘File Tree’ tab. It will show whole repository in the form of a file tree and intended files and directories can be viewed easily. This is the step-wise procedure to traverse a repository by a user.



How to track the changes using Git Extensions?

By clicking the ‘Diff’ tab comparison can be made with respect to the previous commit as shown in the figure.



Note: Newly added lines are marked as '+'sign and shown with green color and the deleted ones as '-' sign and presented with red color.

How to perform Commit & Push?

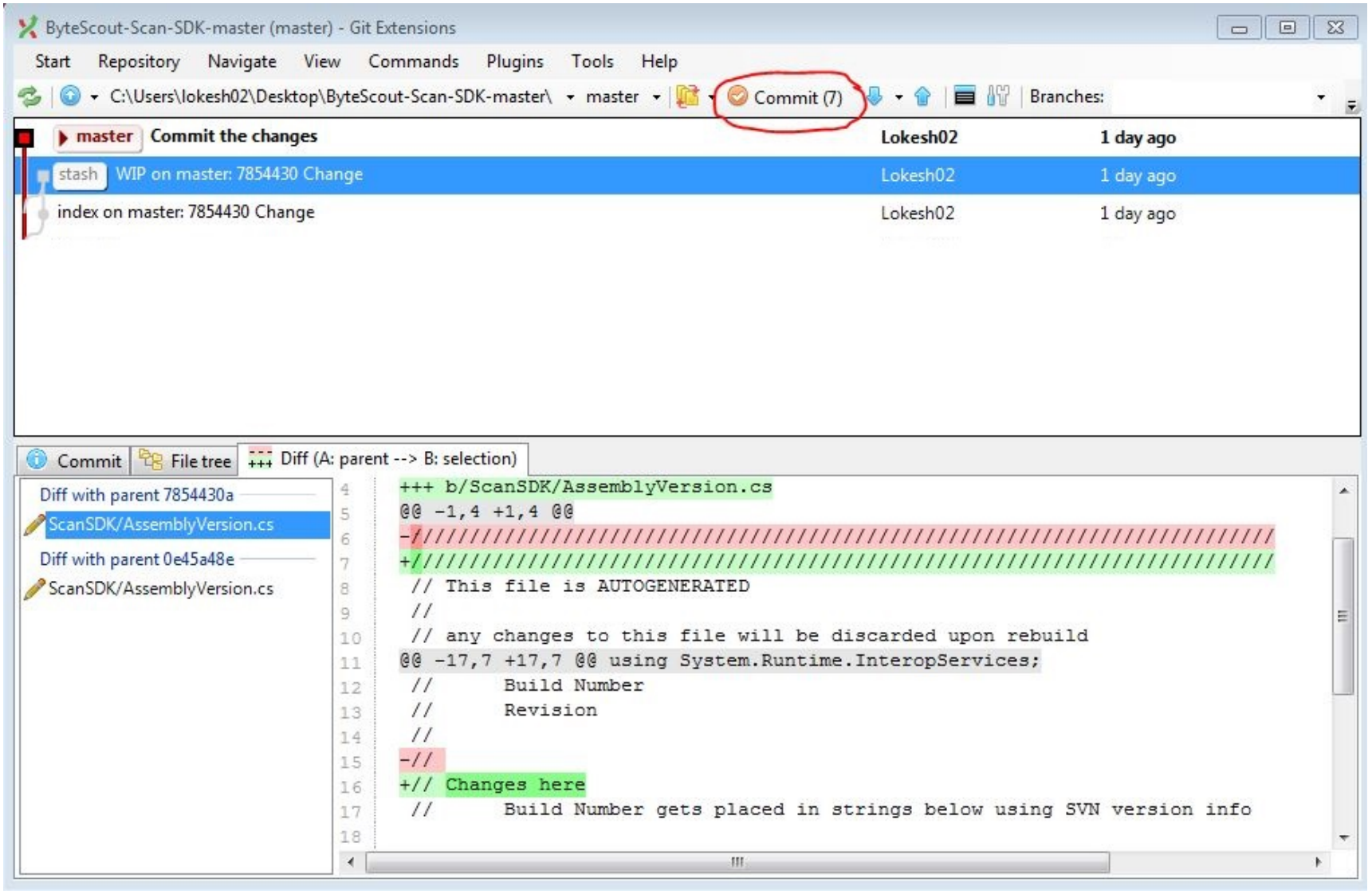
This action of committing and then pushing the given code to a remote repository is divided into two operations:

1. Committing to Local Repository
2. Committing to Remote Repository

Committing to Local Repository with Git Extensions

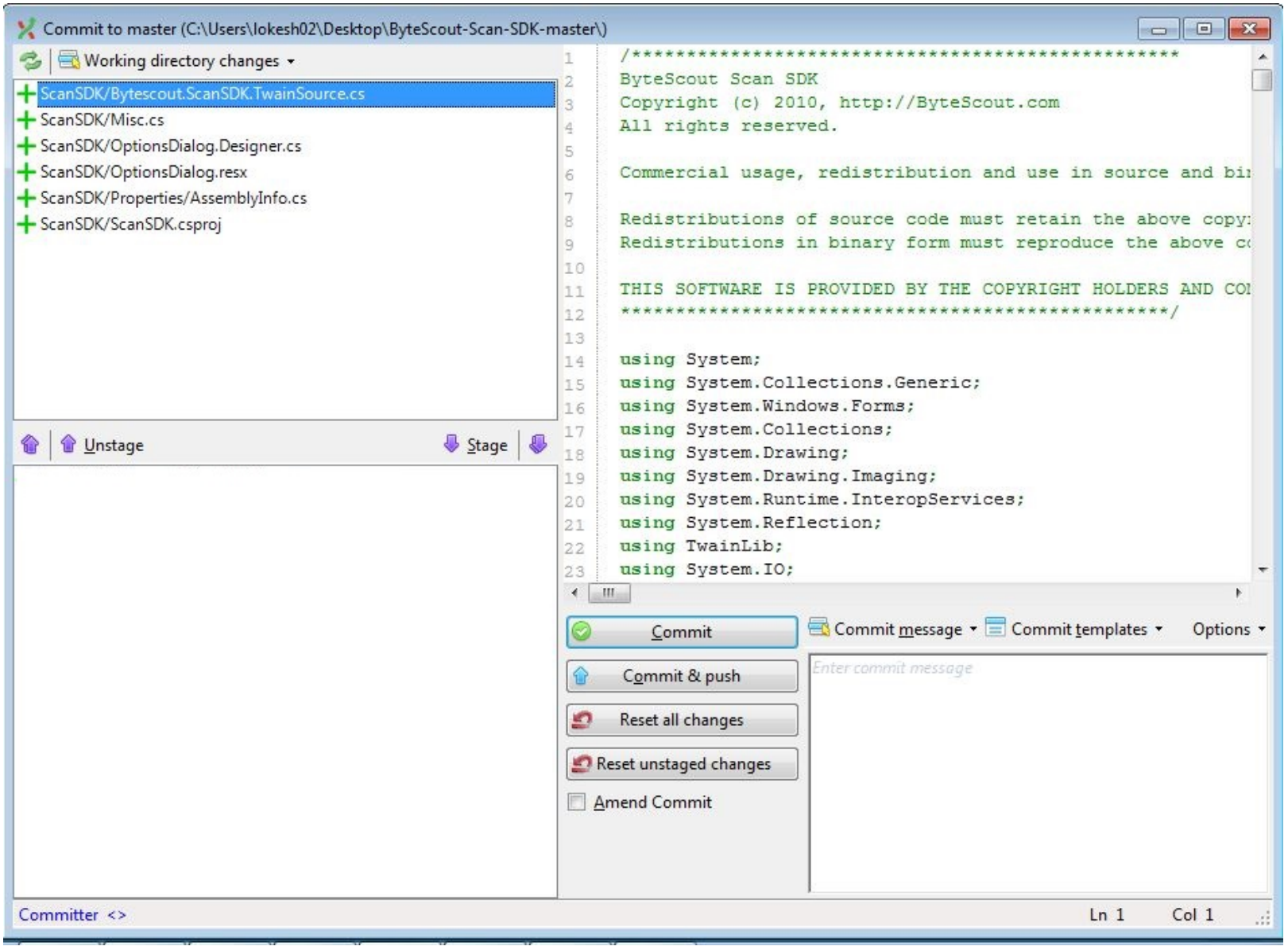
This process of committing involves various steps:

1. To start the process click the 'Commit' icon.



The respective fields are displayed in the newly opened commit window (Refer figure):

- Working directory files.
- Staged files.
- Details of the modifications compared to the respective repo code.
- Input box for the commit message to be entered by the user.



The top left section will show the working directory files which were modified since the last commit:

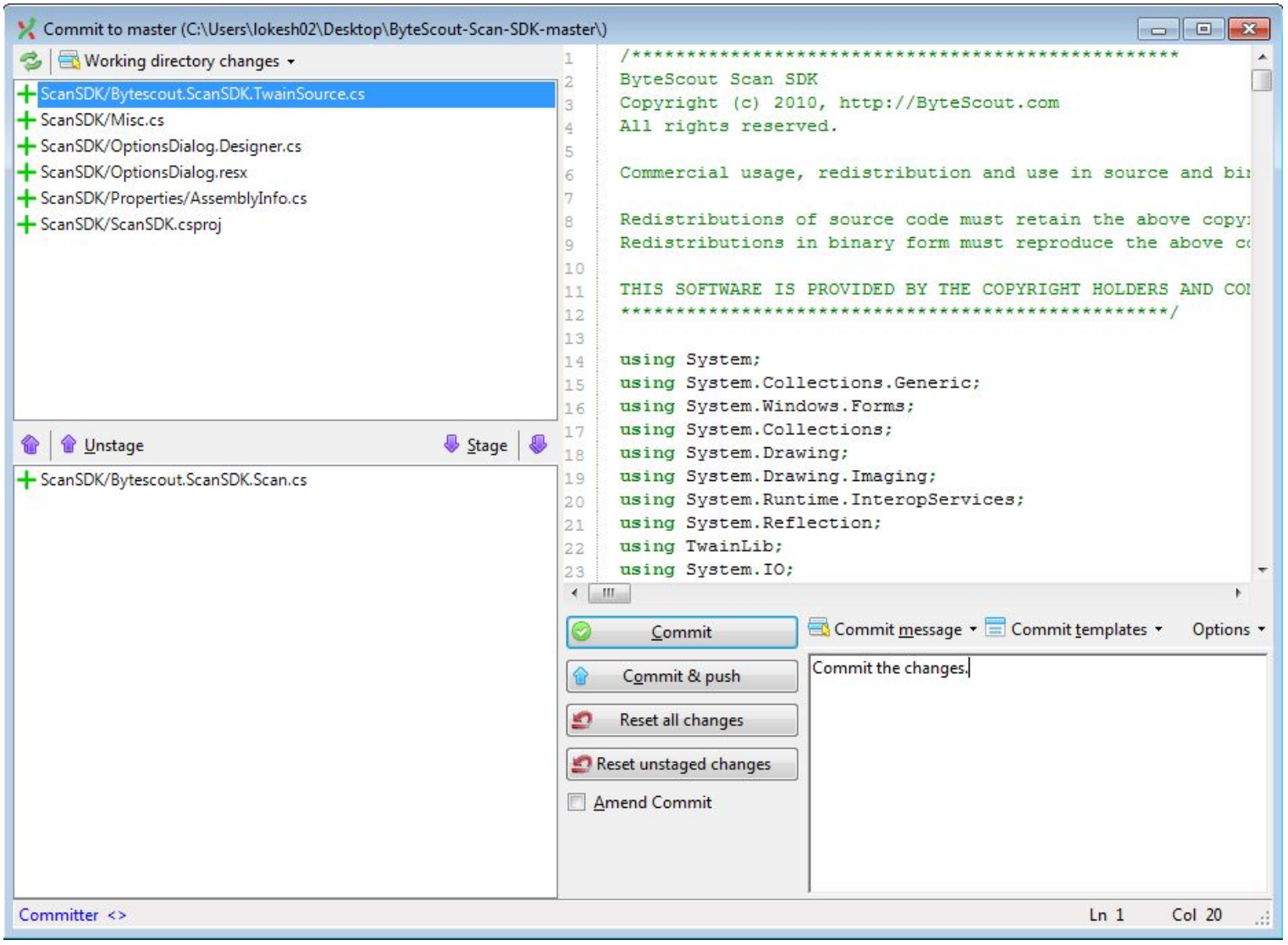
Icon	Meaning

 This represents all existing files that were edited after last commit

 This represents files which are removed after last commit

 This represents new files which are added after last commit

1. Click any file from working directory file list. The files with their modified changes (highlighted in Red and Green) are displayed on right section. This way we can identify file-wise discrete changes.
2. When the changes are verified, files can be staged by using the ‘Stage’ option (Refer Figure). These Staged files are *ready to commit* and provides an easy way for filtering certain files not committed during the previous stage for the users.



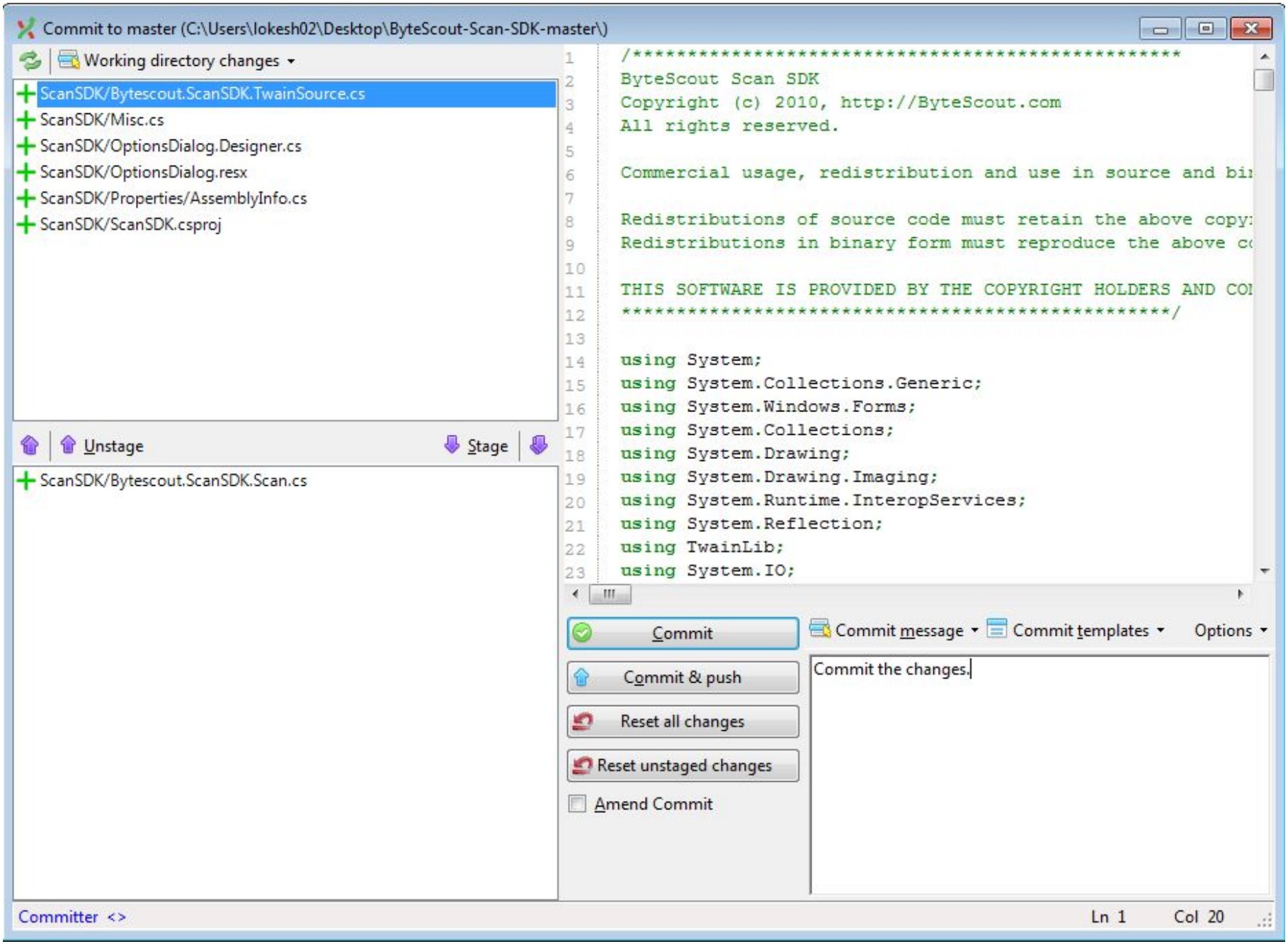
Note: Users can also *Unstage* a staged file by using the 'Unstage' option opposite to *Stage* icon.

3. When the verification and staging of all required files is done, you have to provide a suitable message to show current commit action. Commit History screen will show

this message. The purpose of commit action can be easily interpreted if an appropriate message is given.

2. Commit the files using ‘Commit’ button. Status of the commit operation is displayed in a dialog window. It will also show all the run- time errors of the process, displaying them with the appropriate stage of the whole process.

Note: Clicking ‘Commit’ lets the files to be committed into the local repository and not into the remote repository.

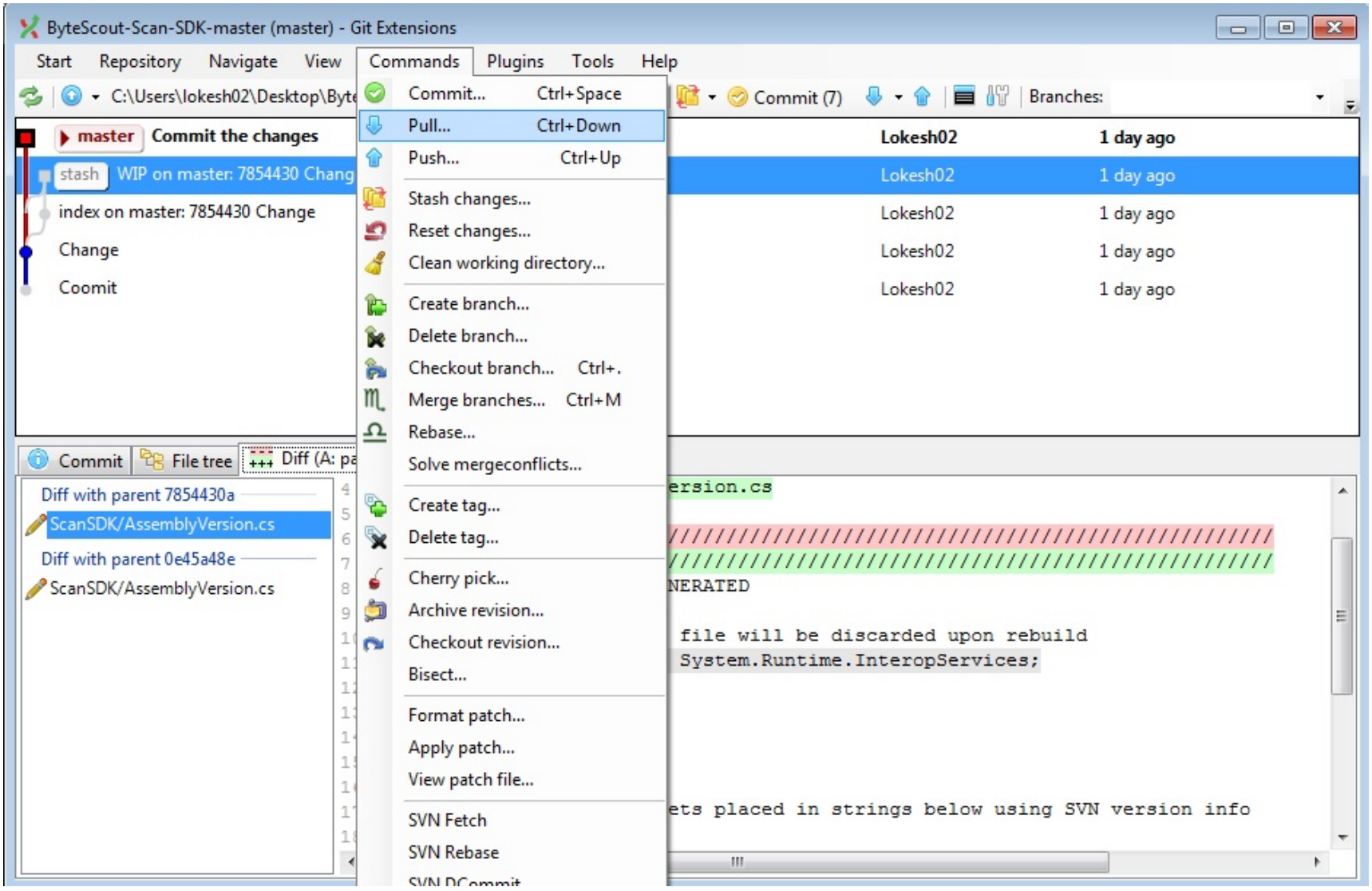

Commit to master (C:\Users\lokesh02\Desktop\ByteScout-Scan-SDK-master\)
Working directory changes
ScanSDK/Bytescout.ScanSDK.TwainSource.cs
ScanSDK/Misc.cs
ScanSDK/OptionsDialog.Designer.cs
ScanSDK/OptionsDialog.resx
ScanSDK/Properties/AssemblyInfo.cs
ScanSDK/ScanSDK.csproj
Unstage
Stage
ScanSDK/Bytescout.ScanSDK.Scan.cs
/**********
ByteScout Scan SDK
Copyright (c) 2010, http://ByteScout.com
All rights reserved.
Commercial usage, redistribution and use in source and bi
Redistributions of source code must retain the above copy
Redistributions in binary form must reproduce the above c
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CO
**********/
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using System.Collections;
using System.Drawing;
using System.Drawing.Imaging;
using System.Runtime.InteropServices;
using System.Reflection;
using TwainLib;
using System.IO;
Commit
Commit message
Commit templates
Options
Commit & push
Commit the changes.
Reset all changes
Reset unstaged changes
Amend Commit
Committer <>
Ln 1
Col 20

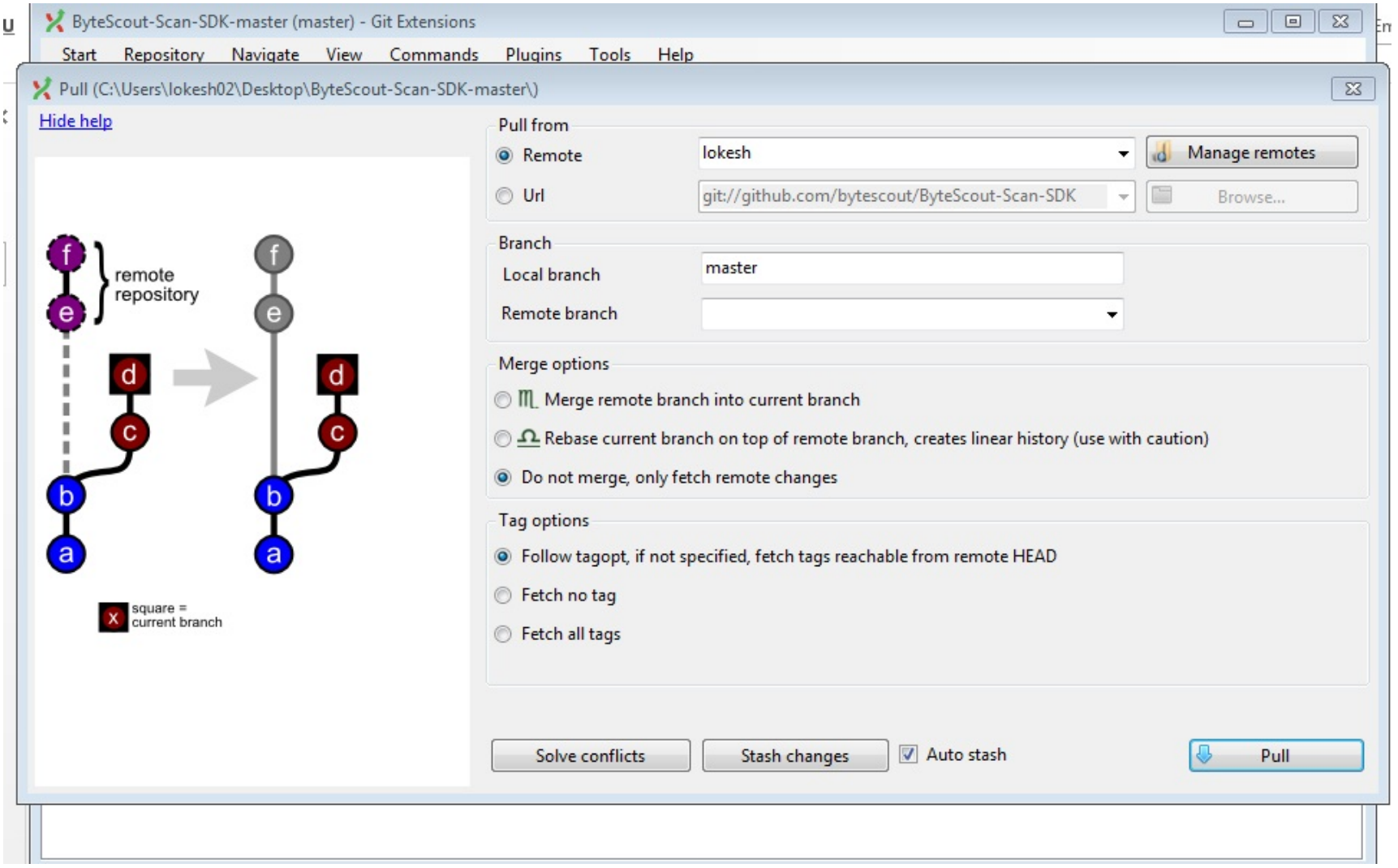
Committing to Remote Repository

‘Push’ action is used to move the files of local repository into remote repository. It needs to be ensured by the user that the code which was taken lastly from the remote repository is not modified before you perform push action. There is every possibility that a repository could have been modified when it has been pulled or cloned by another person. This generally happens in a multi-user environment where a lot of branching and merging happens. Hence, it is necessary for a user to perform a Pull before opting for a Pushing action for the committed code.

Pulling the code

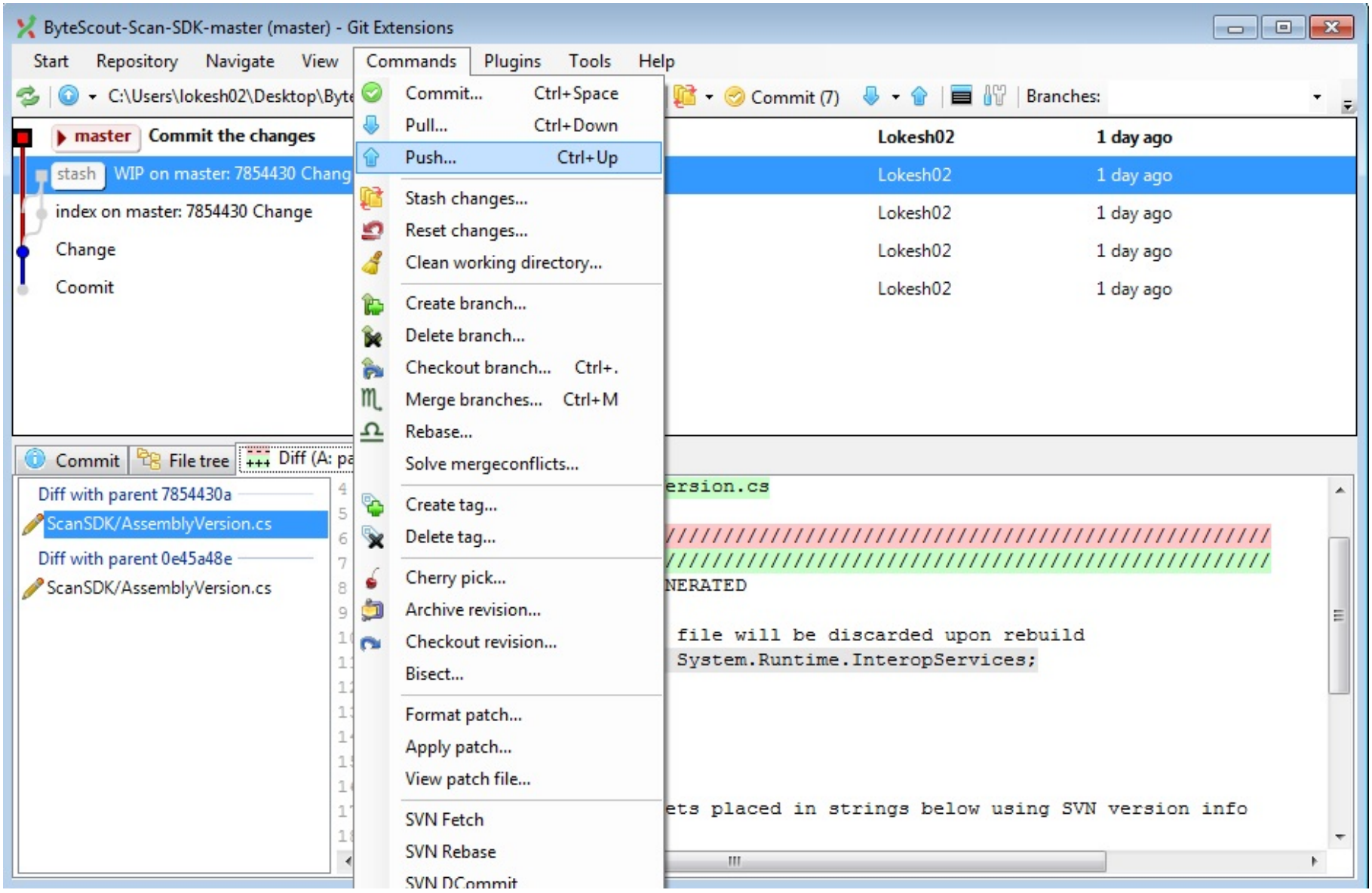
1. Open the pull Window by clicking on the ‘Pull’ icon as shown in the figure. Provide the remote repo URL and select the remote branch form there.
2. It is necessary to select the Do Not Merge Option (*Do not merge, only fetch remote changes*) and ‘Auto Stash’ option.
3. Lastly click the ‘Pull’ button.

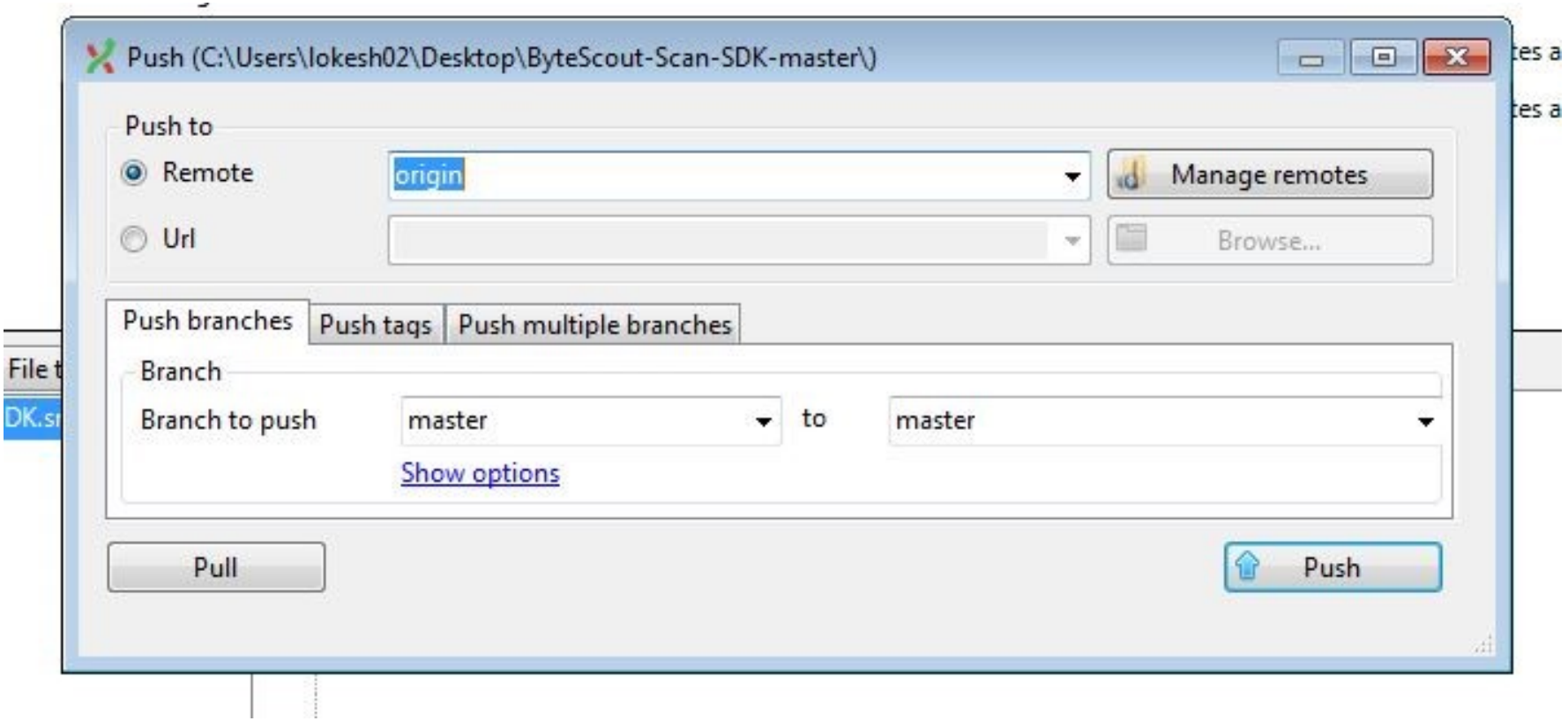

ByteScout-Scan-SDK-master (master) - Git Extensions
Start
Repository
Navigate
View
Commands
Plugins
Tools
Help
Commit... Ctrl+Space
Pull... Ctrl+Down
Push... Ctrl+Up
Stash changes...
Reset changes...
Clean working directory...
Create branch...
Delete branch...
Checkout branch... Ctrl+.
Merge branches... Ctrl+M
Rebase...
Solve mergeconflicts...
Create tag...
Delete tag...
Cherry pick...
Archive revision...
Checkout revision...
Bisect...
Format patch...
Apply patch...
View patch file...
SVN Fetch
SVN Rebase
Commit (7)
Branches:
master
Commit the changes
stash
WIP on master: 7854430 Chang
index on master: 7854430 Change
Change
Coomit
Lokesh02
1 day ago
Commit
File tree
Diff with parent 7854430a
ScanSDK/AssemblyVersion.cs
Diff with parent 0e45a48e
ScanSDK/AssemblyVersion.cs
System.Runtime.InteropServices;



Pushing the code

As of now, the local repository and the remote repository are in sync, so the user now has to click the ‘Push’ button as shown in the figure so that all the locally committed changes can be pushed to the remote repository.


ByteScout-Scan-SDK-master (master) - Git Extensions
Start
Repository
Navigate
View
Commands
Plugins
Tools
Help
Commit... Ctrl+Space
Pull... Ctrl+Down
Push... Ctrl+Up
Stash changes...
Reset changes...
Clean working directory...
Create branch...
Delete branch...
Checkout branch... Ctrl+.
Merge branches... Ctrl+M
Rebase...
Solve mergeconflicts...
Create tag...
Delete tag...
Cherry pick...
Archive revision...
Checkout revision...
Bisect...
Format patch...
Apply patch...
View patch file...
SVN Fetch
SVN Rebase
SVN DCommit
Commit (7)
Branches:
master
Commit the changes
stash
WIP on master: 7854430 Chang
index on master: 7854430 Change
Change
Coomit
Lokesh02
1 day ago
Commit
File tree
Diff (A: pa
Diff with parent 7854430a
ScanSDK/AssemblyVersion.cs
Diff with parent 0e45a48e
ScanSDK/AssemblyVersion.cs
ersion.cs
NERATED
file will be discarded upon rebuild
System.Runtime.InteropServices;
ets placed in strings below using SVN version info



We hope that this tutorial helps you in getting familiarized with the use of Git using the tool Git Extensions.

Tags: Git

Related Posts

Be the First to Learn about our Updates

November 17th, 2017 | Comments Off

Why use ASP.NET for Web Application Development?

October 28th, 2017 | 0 Comments

Ansari Azad • a year ago

Hi, how to change committer picture in git extensions ?
Thanks in advance !!

Reply • Share ›

ALSO ON **BYTESCOUT BLOG**

How to setup a Windows Server with ASP.NET MVC App on AWS EC2

4 comments • a year ago

Debbie33499 — We have created 240+ applications in vb.net using asp.net just using a text-editor. (Over the past 11 years.)Can we just

Big Data: What it is & what it is not (Part 1)

1 comment • a year ago

Dereck Gligorijevic — Yeah it is much easier to understand the whole Big Data concept once you introduce these 3 Vs.

Microsoft Bot Framework chat bot experience comparing to Telegram bot

1 comment • 2 years ago

Rachel Lancaster — Nice article

ByteScout Blog: Top 7 Image Editing Apps for Android

1 comment • 3 years ago

Jeni — I want to add here another great editor for the android users. Its Adobe Lightroom for Android 1.4 . Now its available for android and

PURCHASE

- Suites
- Single Products
- Local Resellers

DOWNLOAD

- Trial Versions
- Freeware Utilities

COMPANY

- SOLUTIONS
- Contact Us
- About Us
- Partnership