Computer Programming Fundamentals

kangwon.lee@kpu.ac.kr

2014 Spring Semester

GIT

&

TortoiseGIT

- Source Revision Control System
- Open Source
- Developed by Linus Torvalds
- Used to develop Linux

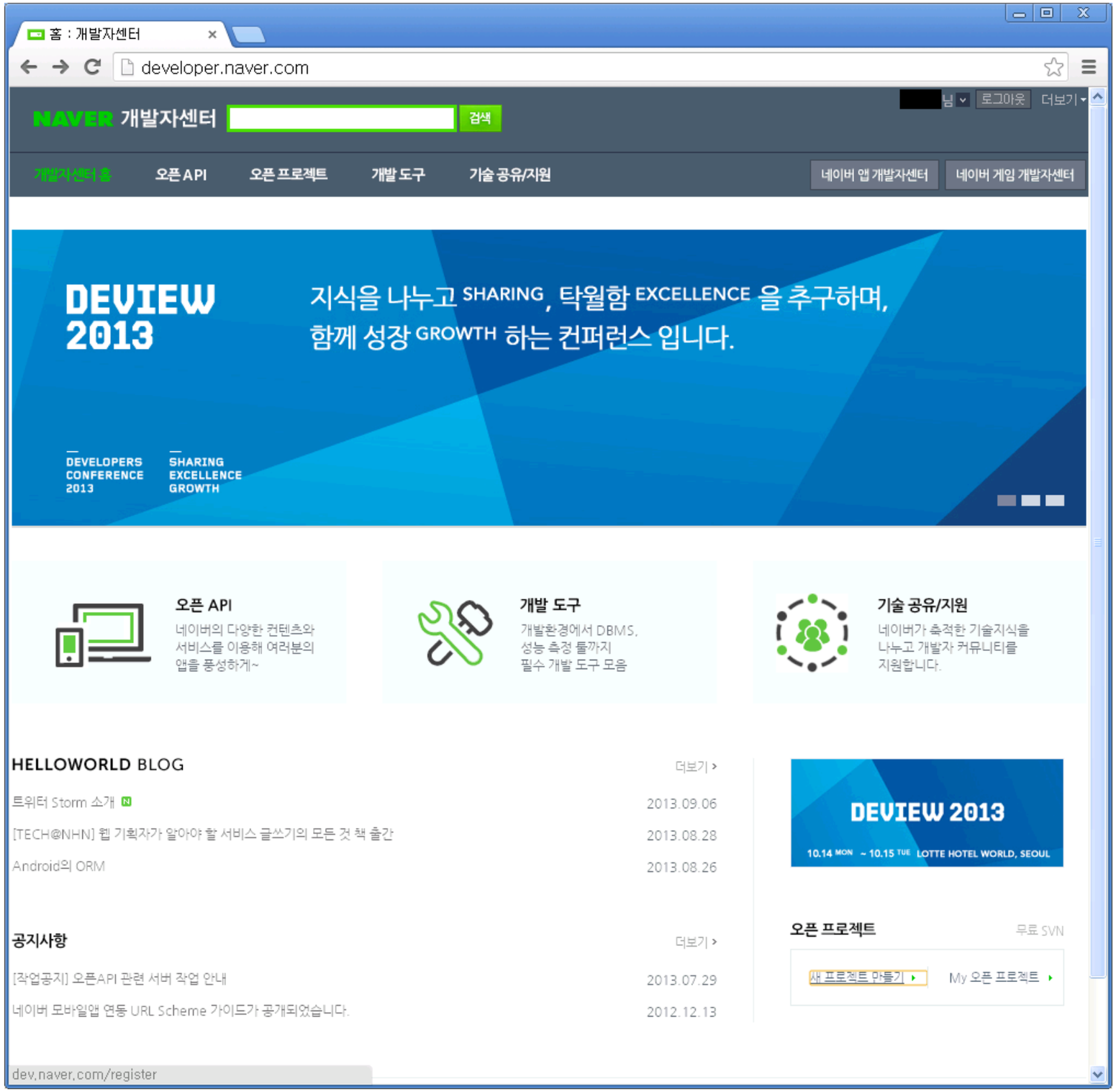
What is GIT?

- Revision Control Systems keeps track of changes in as you modify your source code.
- One day, your code that used to work fine may start acting differently.
- You may want to
  - go back to your source code that worked fine.
  - or analyze which modification caused that problem.
- In any cases, Revision Control Systems may help you.

Why use Version Control?

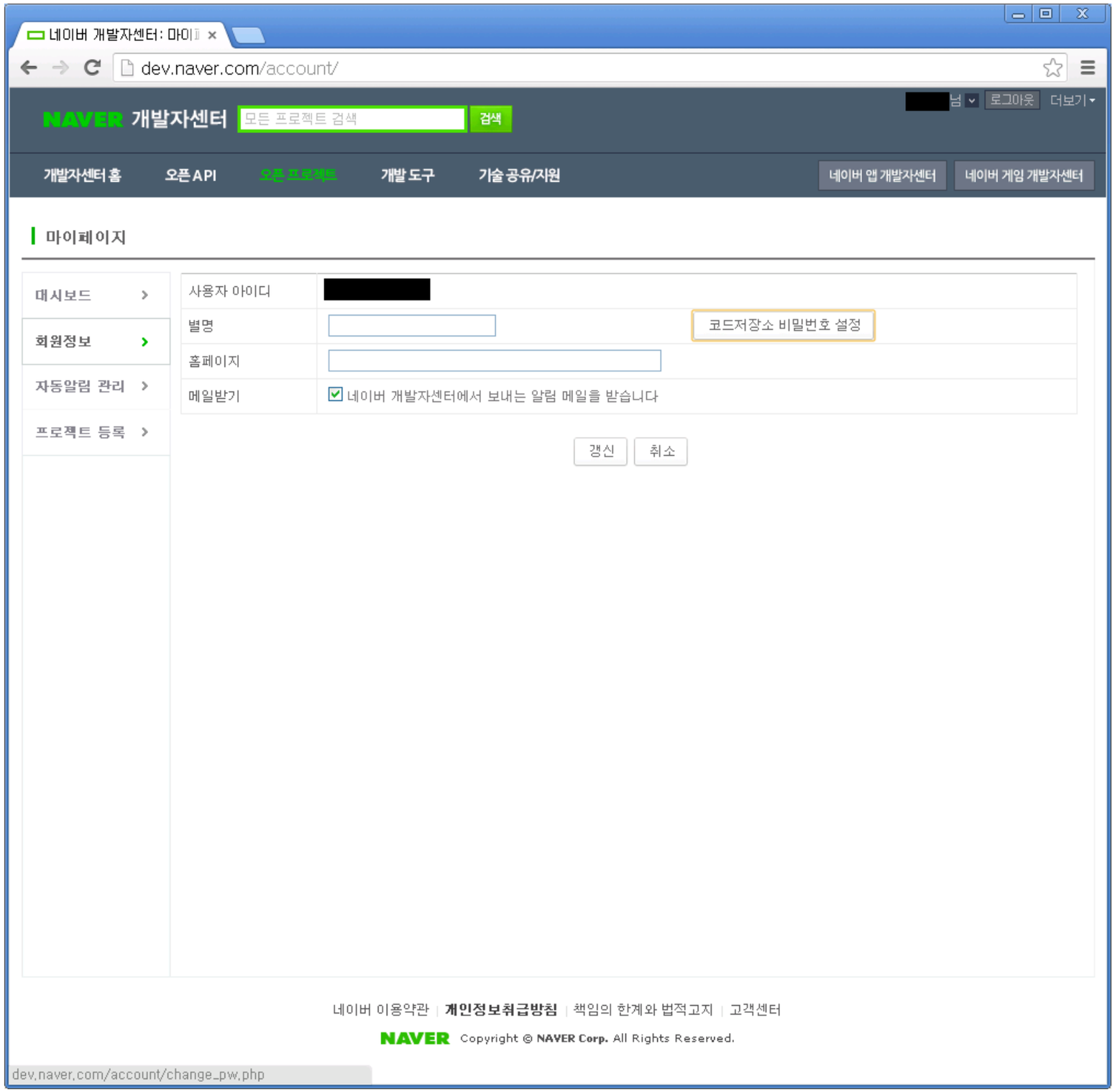
- Log-in to http://www.naver.com
- Create your project in http://dev.naver.com/(10 min)
- Create code repository password @ “My Page” (10 min)
- Look into your “code repository”
  - What do you see?
- Create a new folder “ex01-git”
- Clone your new folder to your PC @ D:\????\[proj ID]
- Make a new file in the folder in your PC
- Add the new file
- Commit
- Modify the new file
- Diff with previous version
- Commit again
- Push to remote repository

Make GIT Repository



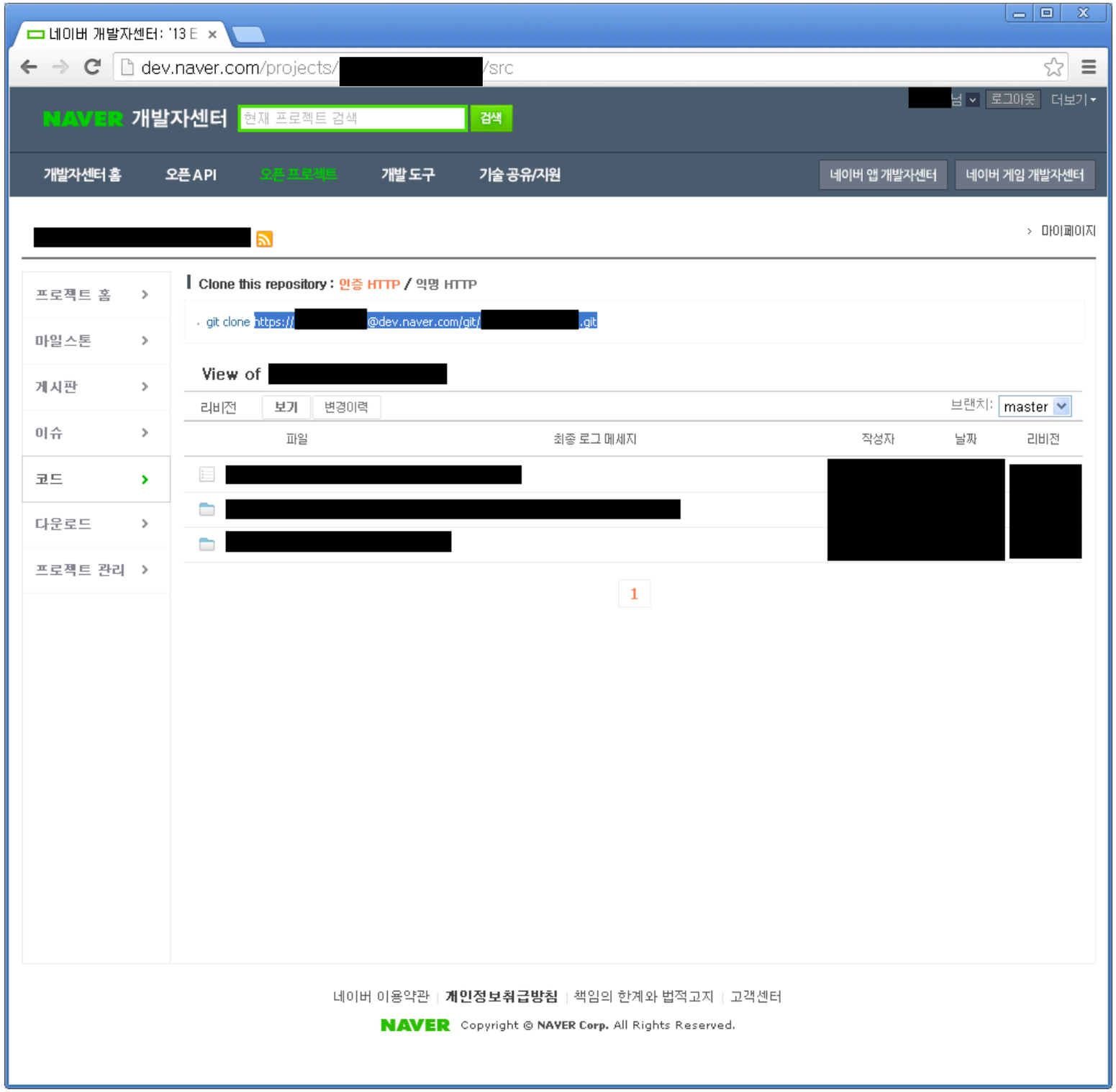
Log-in to http://dev.naver.com

Create your project

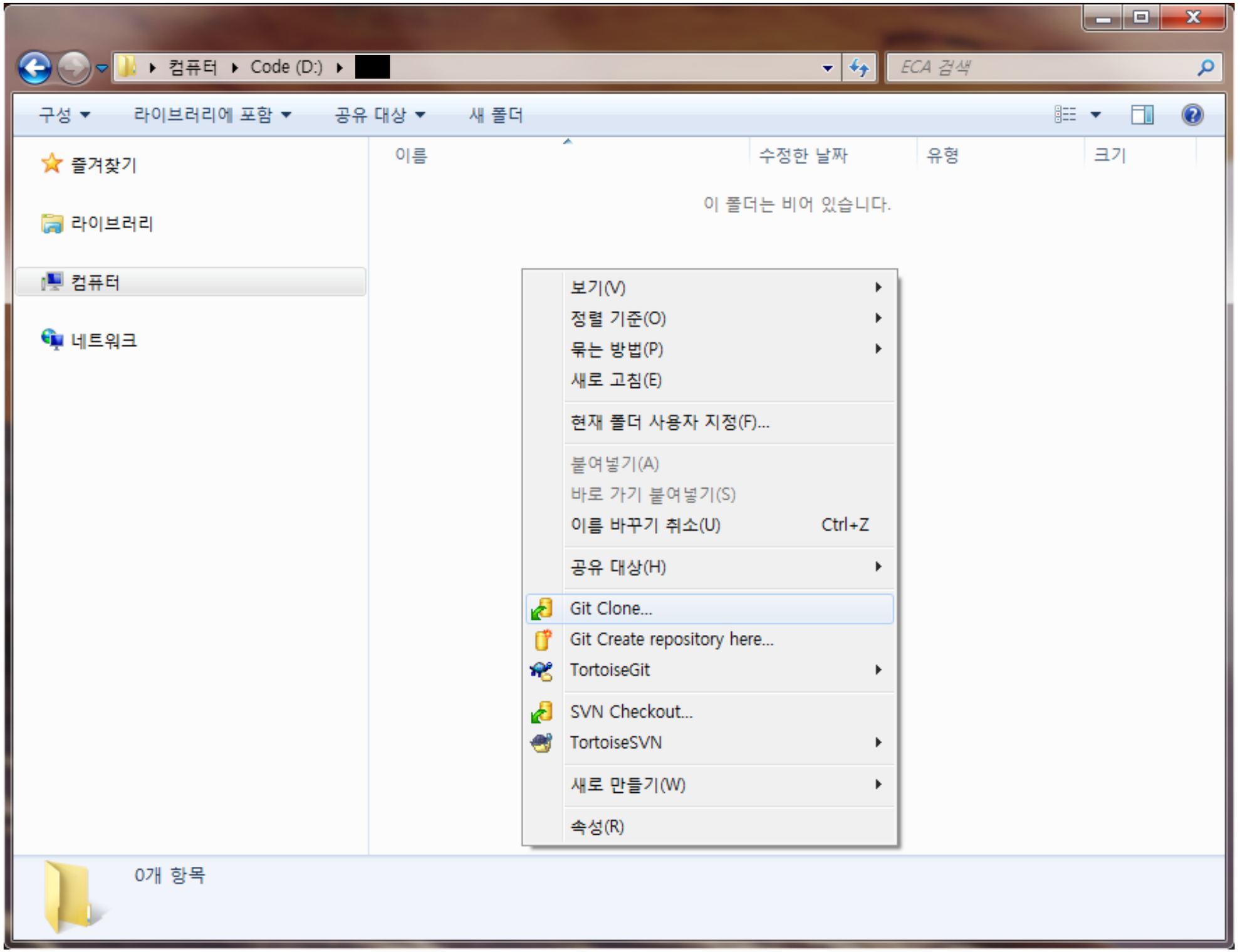


Create code repository password

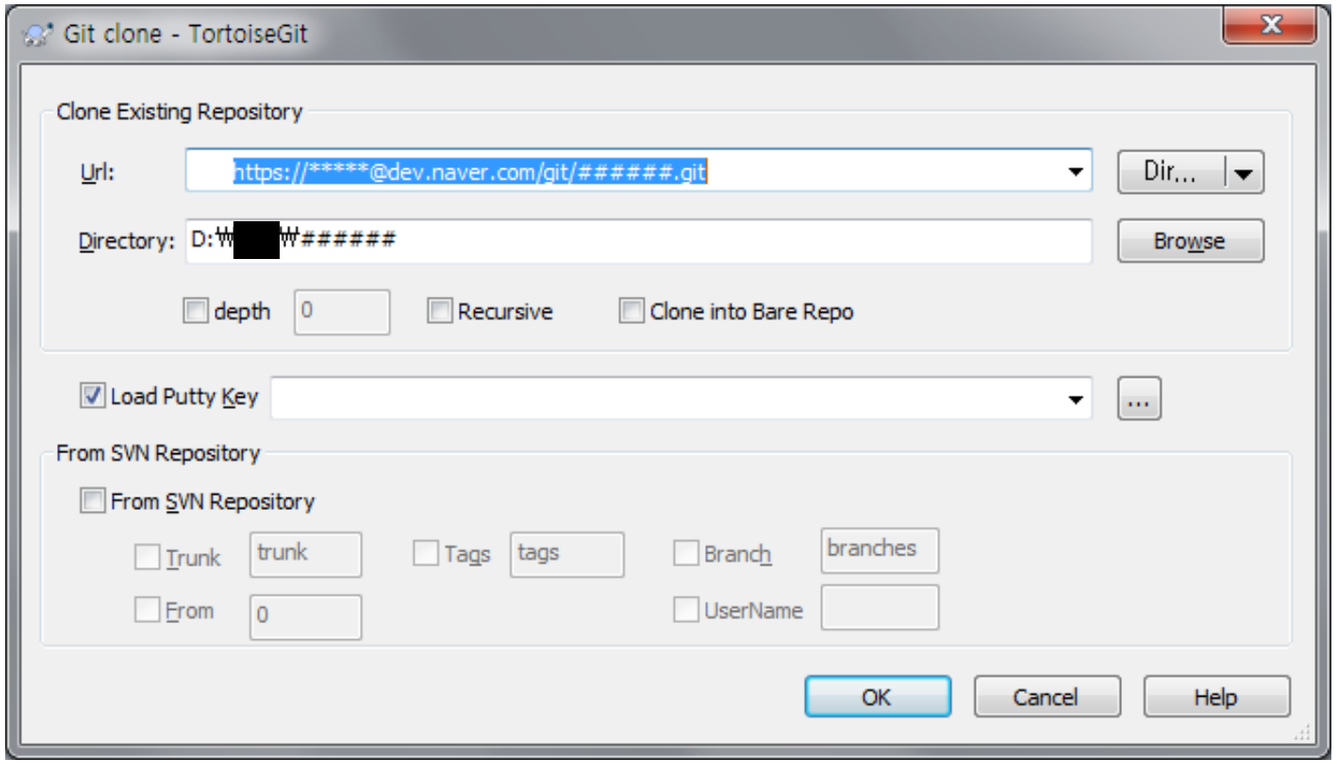
Look into your new project

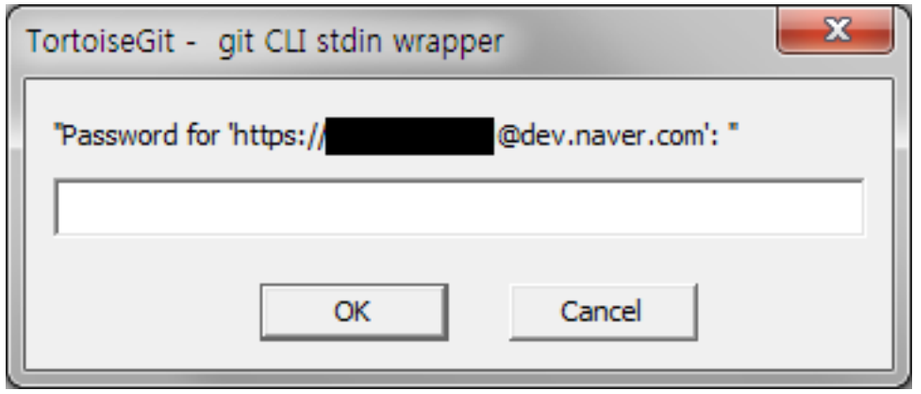


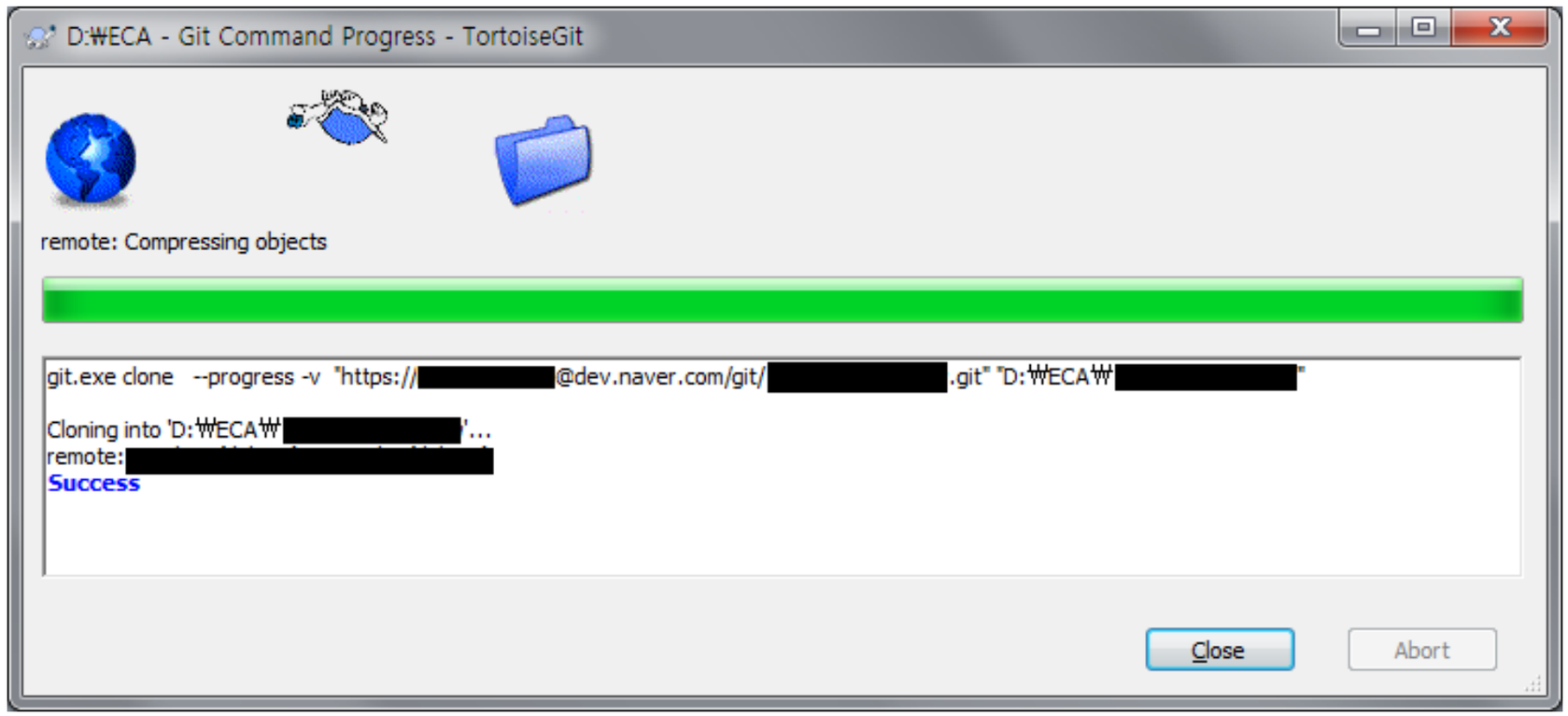
Look into your “code repository”



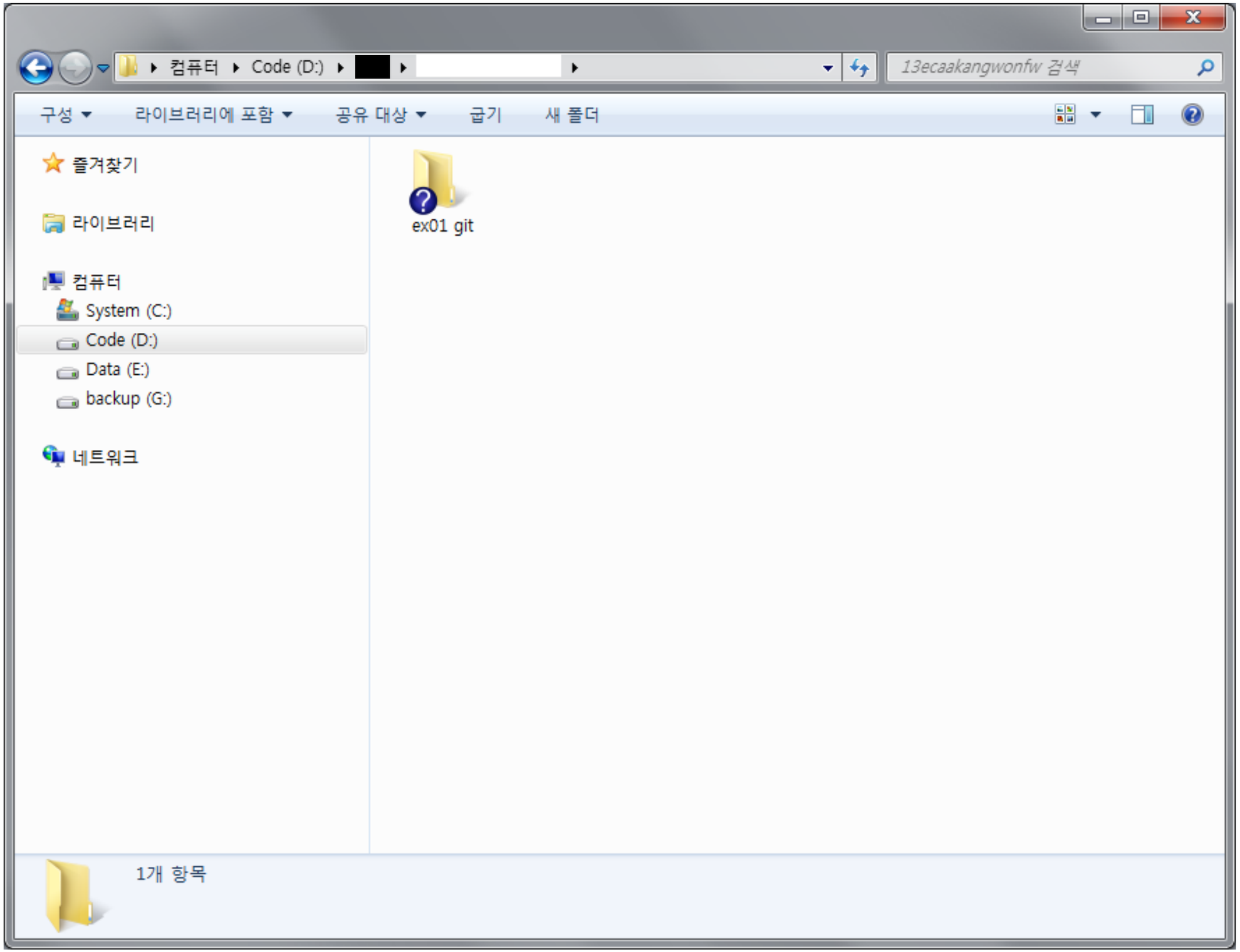
New folder: D:\ECA


Git clone - TortoiseGit
Clone Existing Repository
Url:
https://*****@dev.naver.com/git/######.git
Dir...
Directory:
D:₩[redacted]₩######
Browse
depth
0
Recursive
Clone into Bare Repo
Load Putty Key
...
From SVN Repository
From SVN Repository
Trunk
trunk
Tags
tags
Branch
branches
From
0
UserName
OK
Cancel
Help

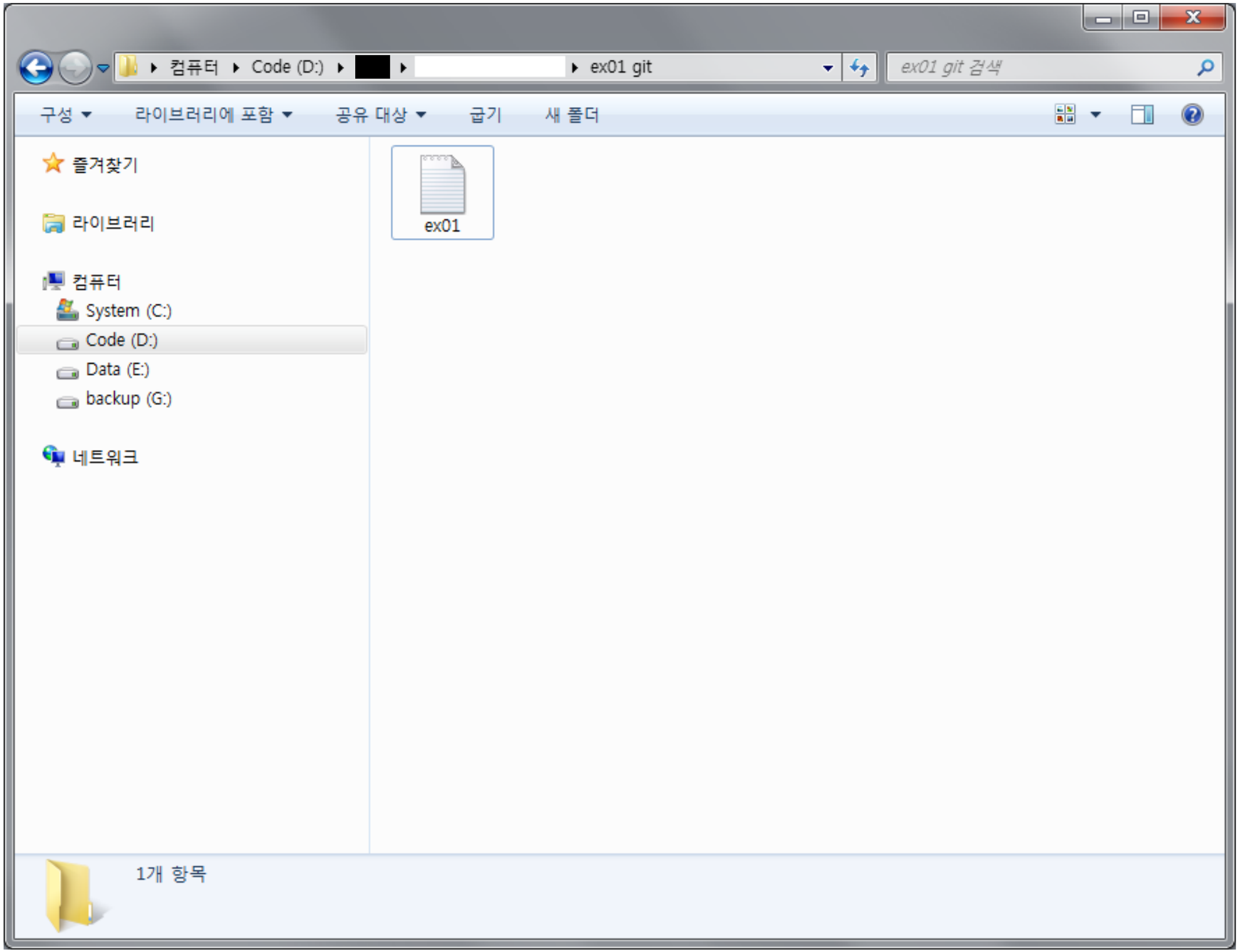

TortoiseGit - git CLI stdin wrapper
'Password for 'https://[redacted]@dev.naver.com': '
OK
Cancel


D:₩ECA - Git Command Progress - TortoiseGit
remote: Compressing objects
git.exe clone --progress -v "https://[redacted]@dev.naver.com/git/[redacted].git" "D:₩ECA₩[redacted]"
Cloning into 'D:₩ECA₩[redacted]'...
remote:[redacted]
Success
Close
Abort

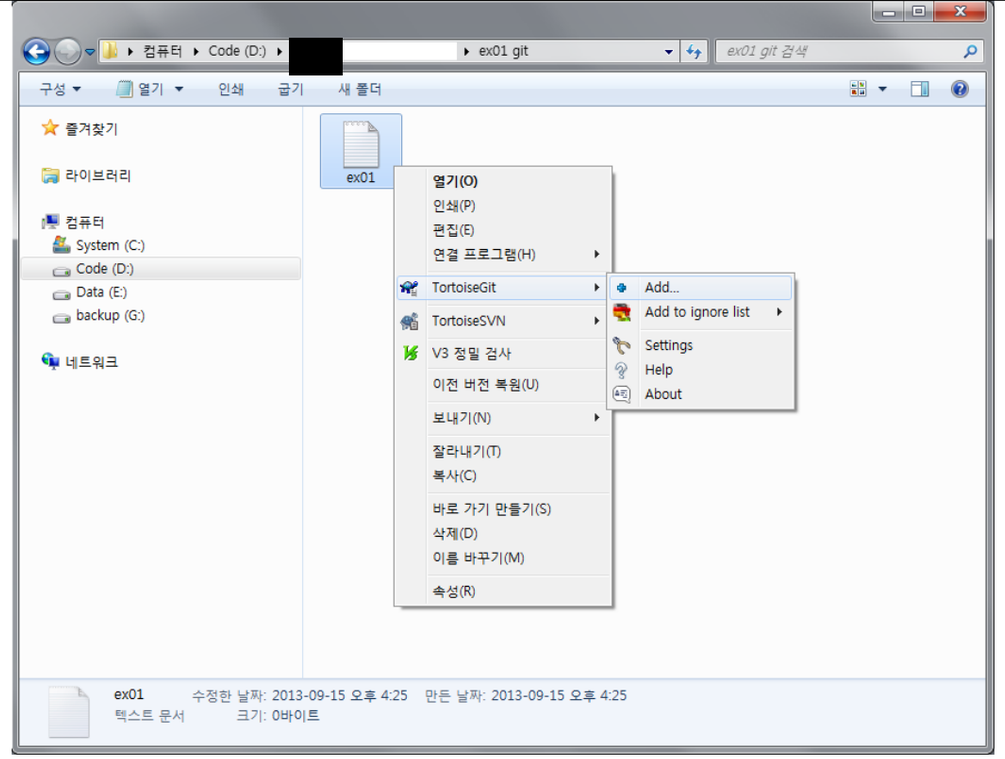
Clone

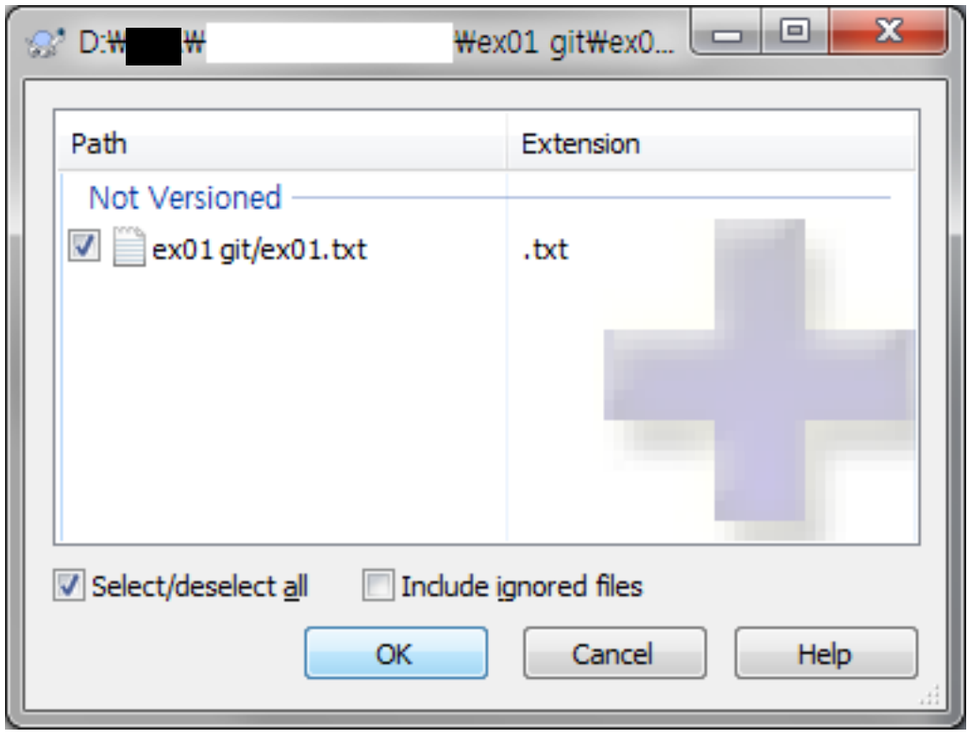


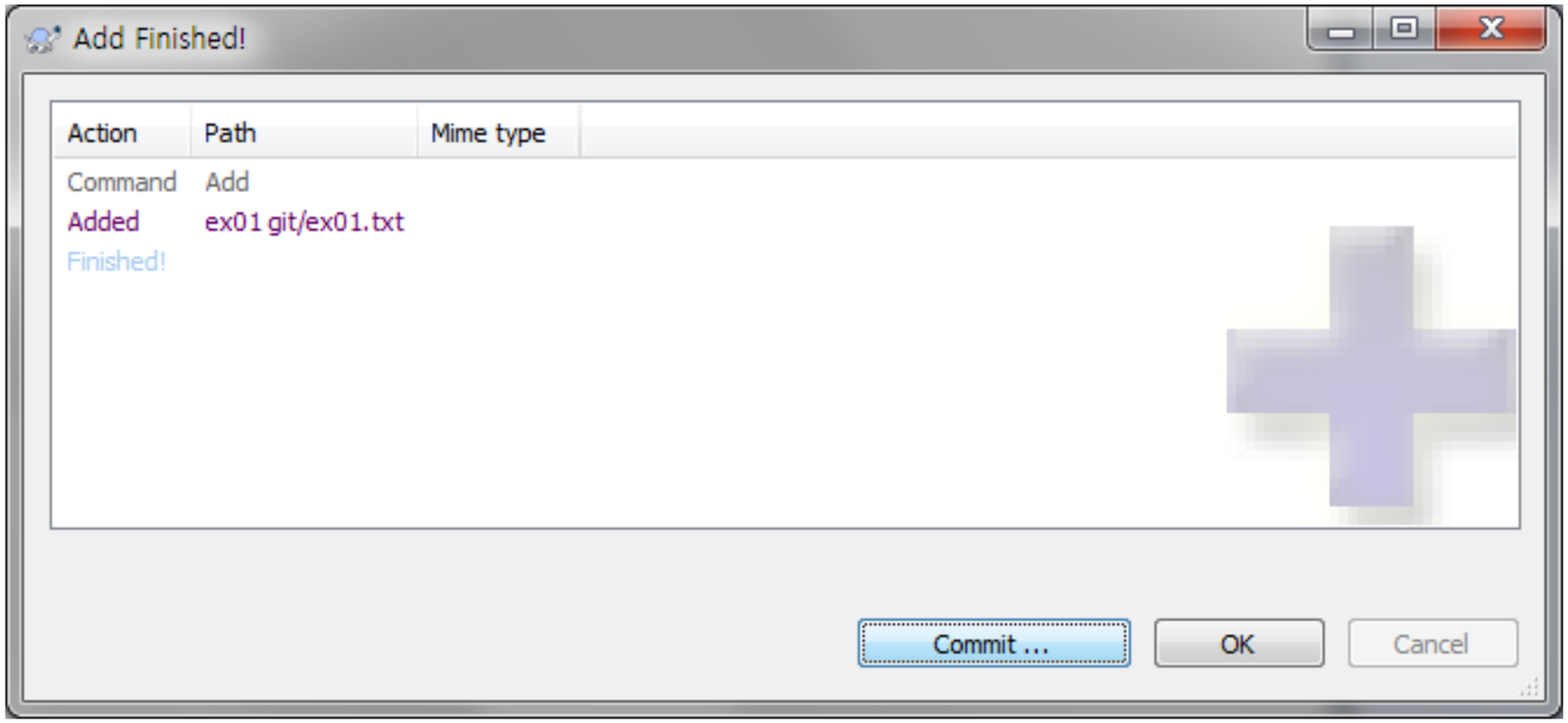
Create a new folder “ex01-git”



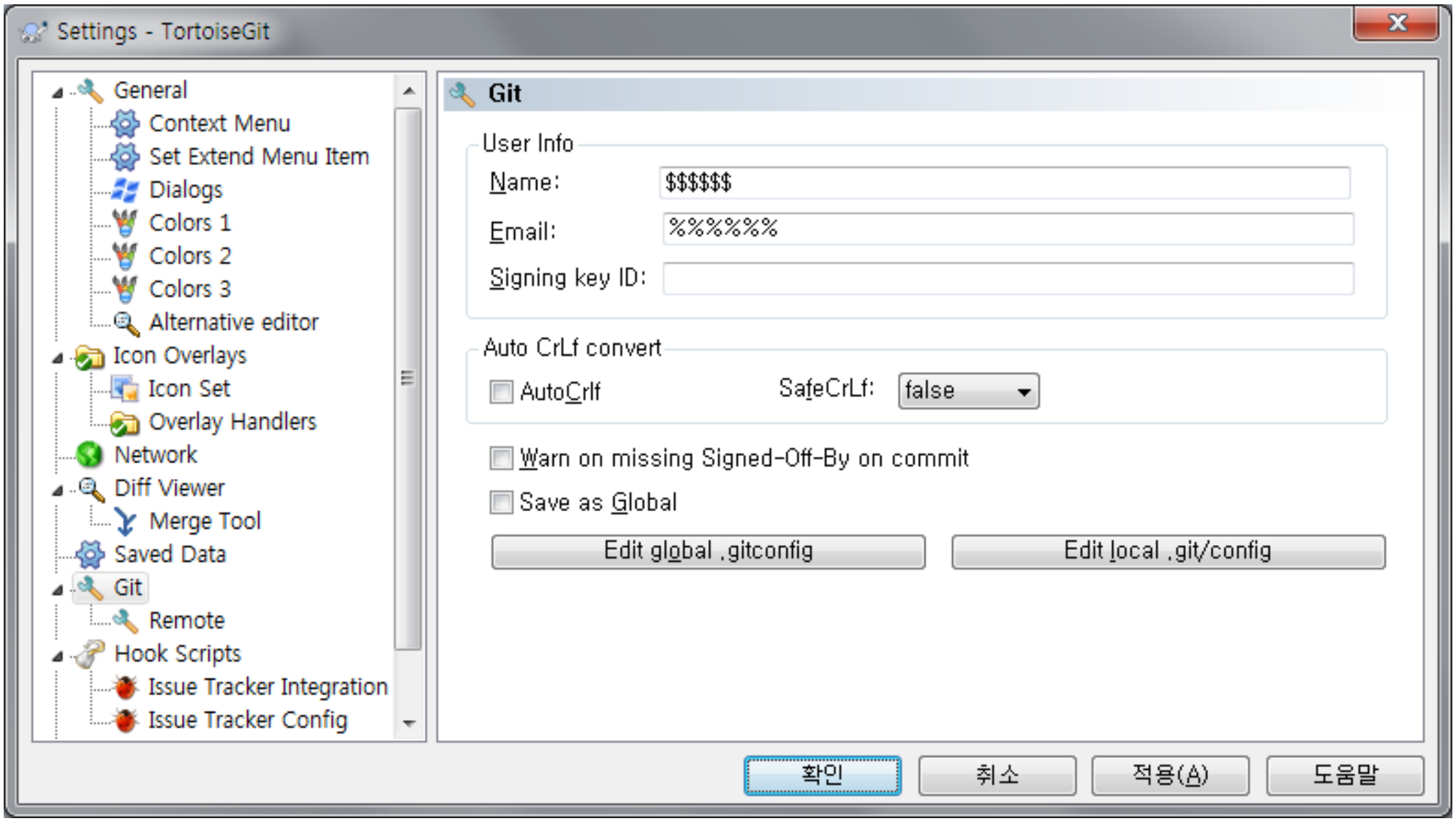
Make a new file



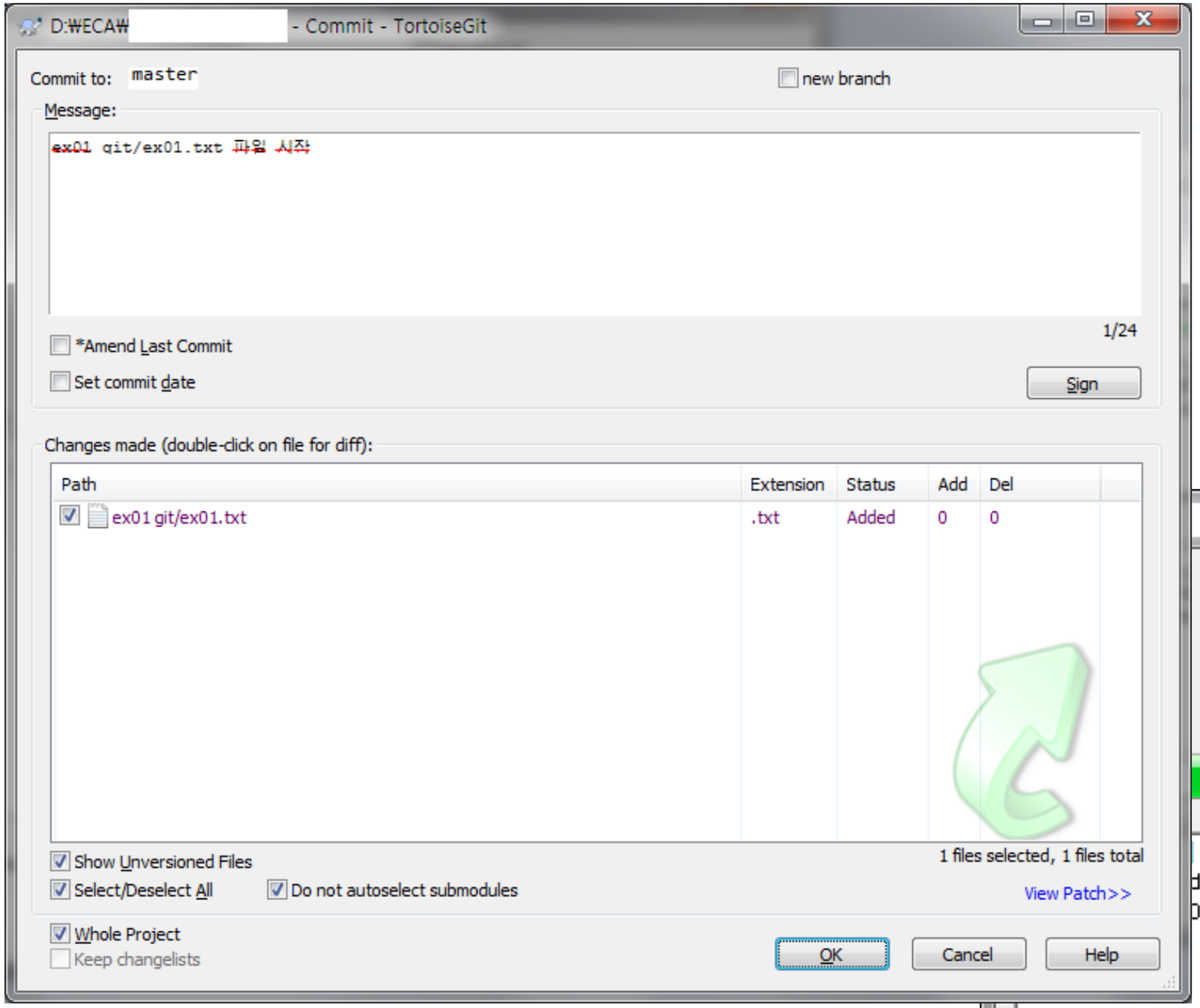


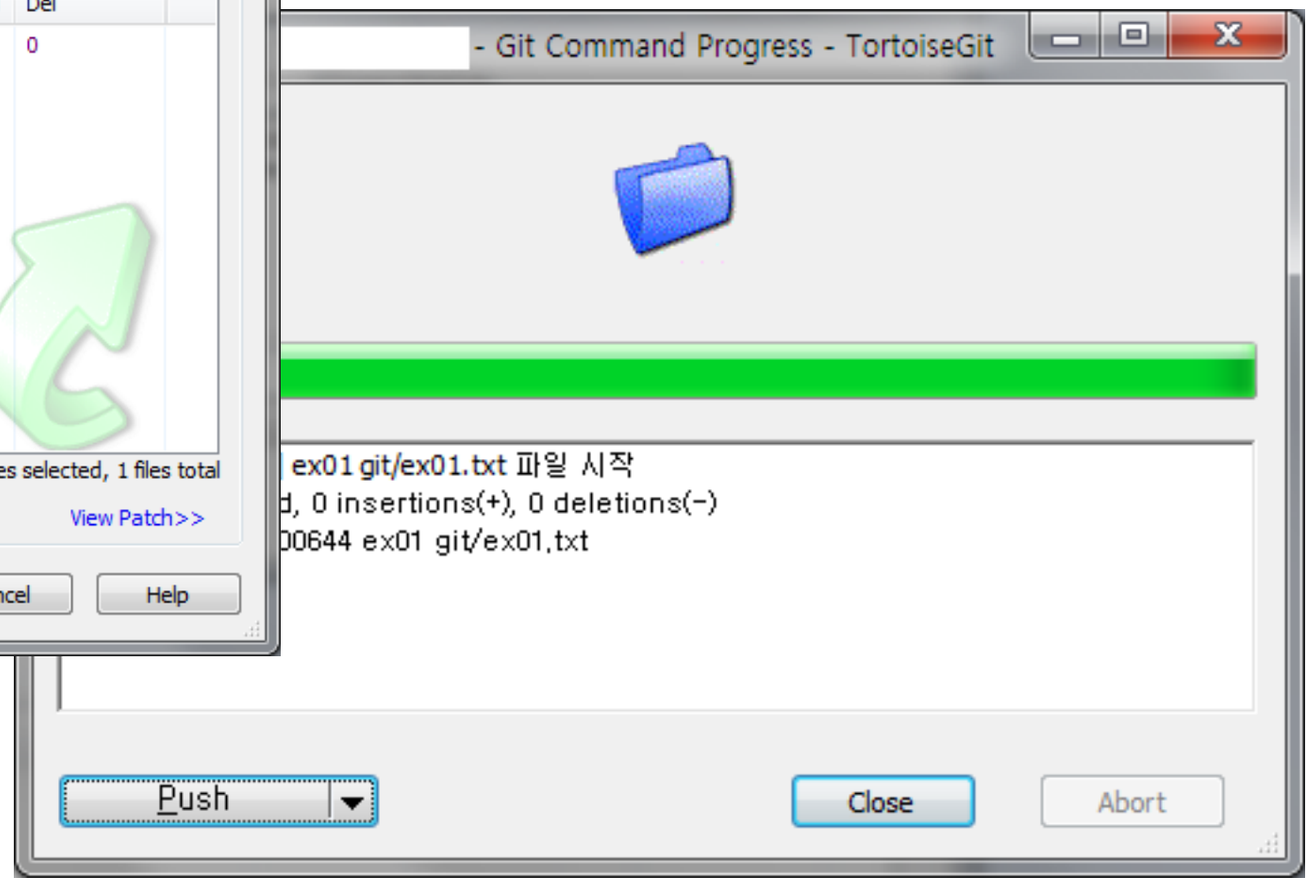


Add the new file



Set name & email

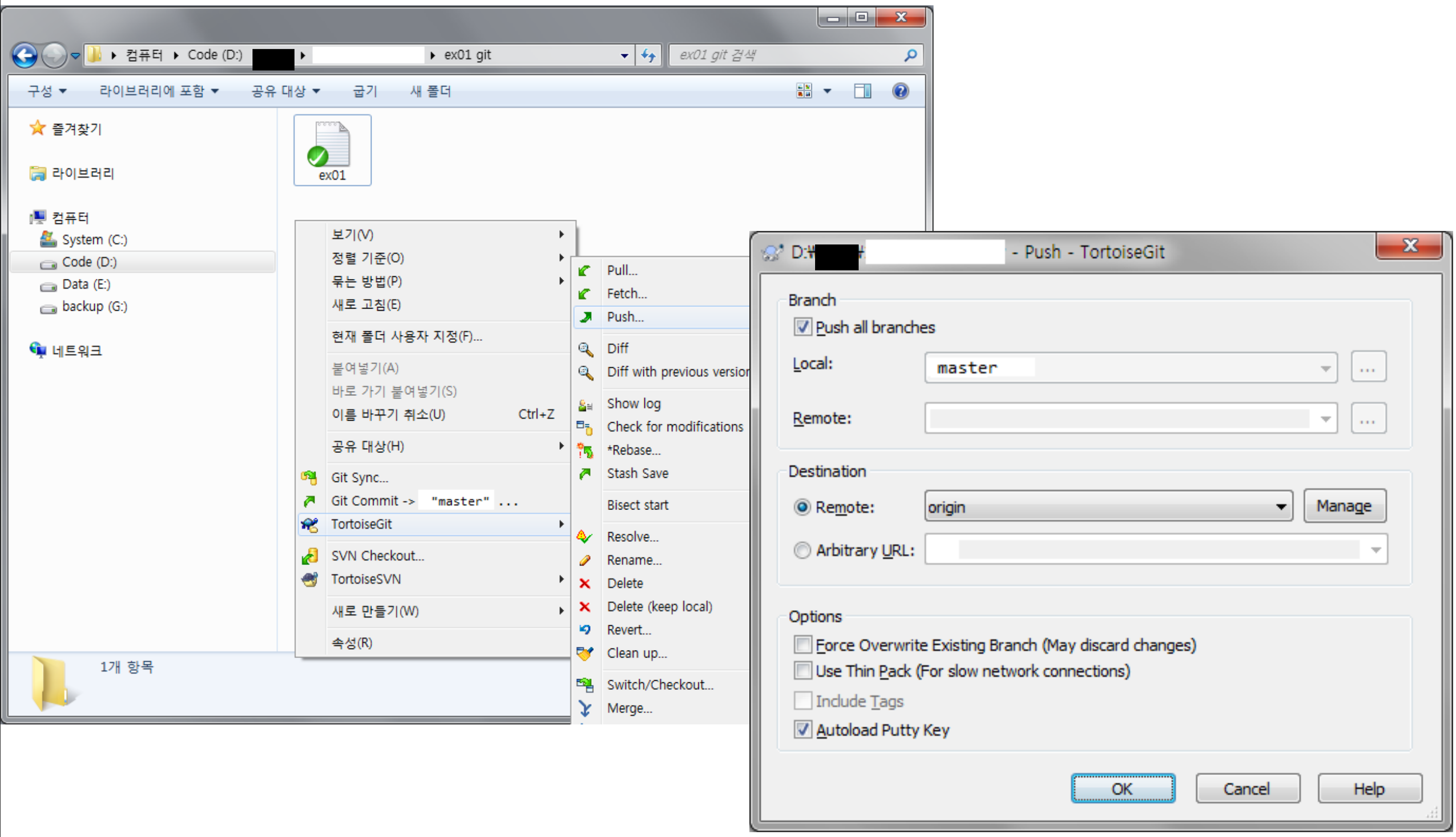

D:₩ECA₩ - Commit - TortoiseGit
Commit to: master
new branch
Message:
ex01 git/ex01.txt 파일 시작
1/24
*Amend Last Commit
Set commit date
Sign
Changes made (double-click on file for diff):
Path
Extension
Status
Add
Del
ex01 git/ex01.txt
.txt
Added
0
0
Show Unversioned Files
1 files selected, 1 files total
Select/Deselect All
Do not autoselect submodules
View Patch>>
Whole Project
Keep changelists
OK
Cancel
Help


- Git Command Progress - TortoiseGit
ex01 git/ex01.txt 파일 시작
d, 0 insertions(+), 0 deletions(-)
00644 ex01 git/ex01.txt
Push
Close
Abort

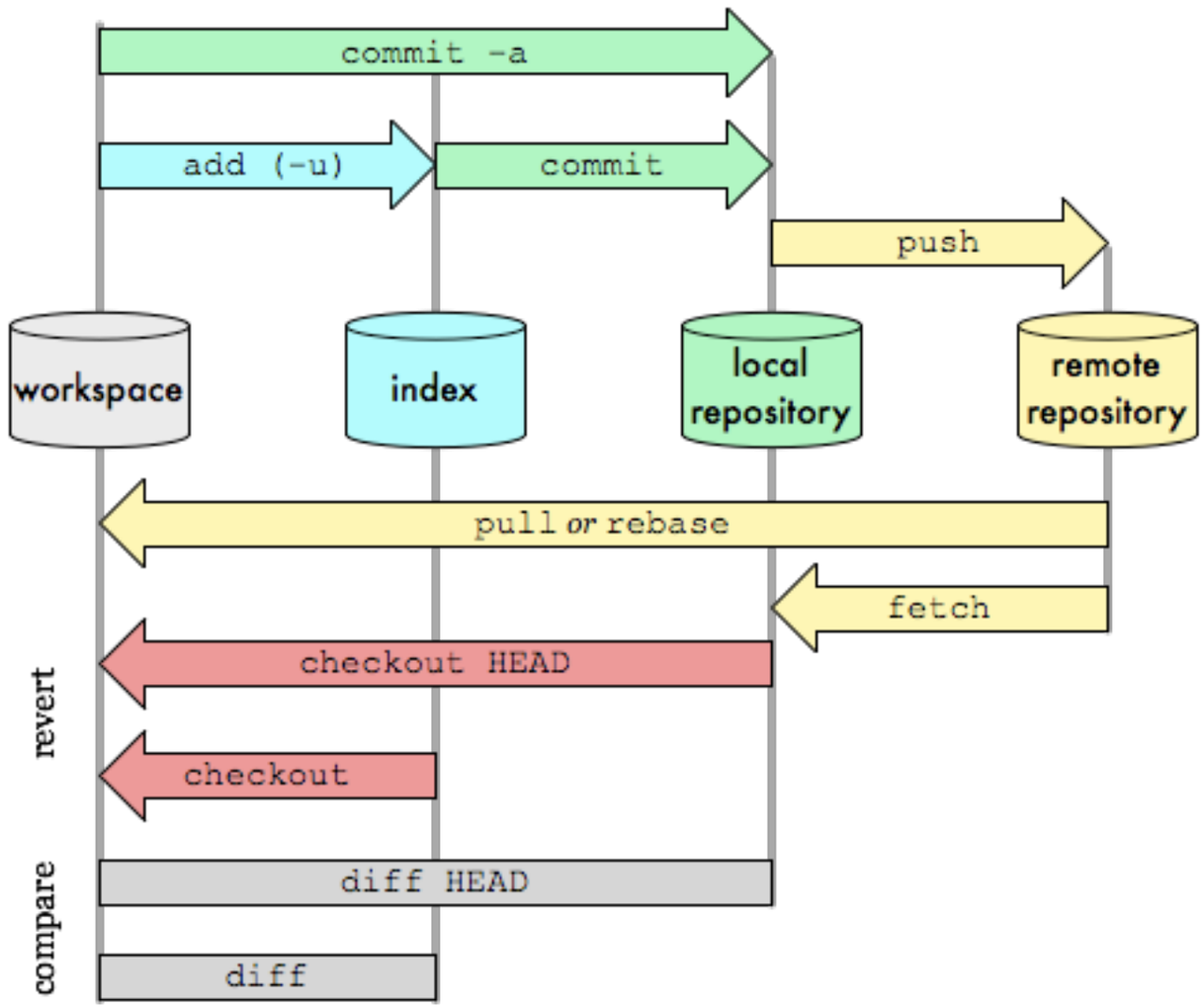
Commit

- Make more files
- Add & Commit the new files
- Modify the new files
- Diff with previous version
- Commit again
- Show log

Try these


컴퓨터 ▸ Code (D:) [redacted] ▸ ex01 git
ex01 git 검색
구성
라이브러리에 포함
공유 대상
굽기
새 폴더
즐겨찾기
라이브러리
컴퓨터
System (C:)
Code (D:)
Data (E:)
backup (G:)
네트워크
ex01
보기(V)
정렬 기준(O)
묶는 방법(P)
새로 고침(E)
현재 폴더 사용자 지정(F)...
붙여넣기(A)
바로 가기 붙여넣기(S)
이름 바꾸기 취소(U) Ctrl+Z
공유 대상(H)
Git Sync...
Git Commit -> "master" ...
TortoiseGit
SVN Checkout...
TortoiseSVN
새로 만들기(W)
속성(R)
1개 항목
Pull...
Fetch...
Push...
Diff
Diff with previous version
Show log
Check for modifications
*Rebase...
Stash Save
Bisect start
Resolve...
Rename...
Delete
Delete (keep local)
Revert...
Clean up...
Switch/Checkout...
Merge...
D:₩[redacted]₩ - Push - TortoiseGit
Branch
Push all branches
Local: master
Remote:
Destination
Remote: origin
Manage
Arbitrary URL:
Options
Force Overwrite Existing Branch (May discard changes)
Use Thin Pack (For slow network connections)
Include Tags
Autoload Putty Key
OK
Cancel
Help

Push



{Fetch, Pull}, {Commit, Push}

- Compare Local & Remote Logs
  - Otherwise, Push
- Delete D:\CPF?\[Project ID]\
  - ? = Class (A or B or C)

Before You Leave