Massachusetts Institute of Technology

MISTI GTL Italy 2015
ITIS Cardano, Pavia

Version Control:
git

Outline

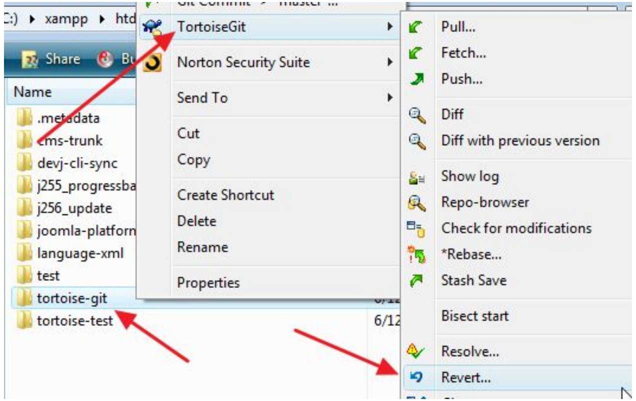
Today:

- undo local changes and commits
- show log
- work with remote repo using github.com

Undo local changes: revert

To undo the changes you haven’t added and committed yet:

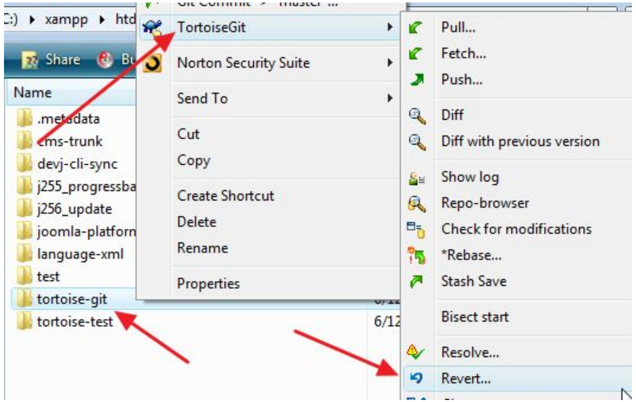
- Right click on the file
- Select TortoiseGit -> Revert
- Select the file that was modified and needs to be restored



Undo local changes: revert

To undo the changes you haven’t added and committed yet:

- Right click on the file
- Select TortoiseGit -> Revert
- Select the file that was modified and needs to be restored

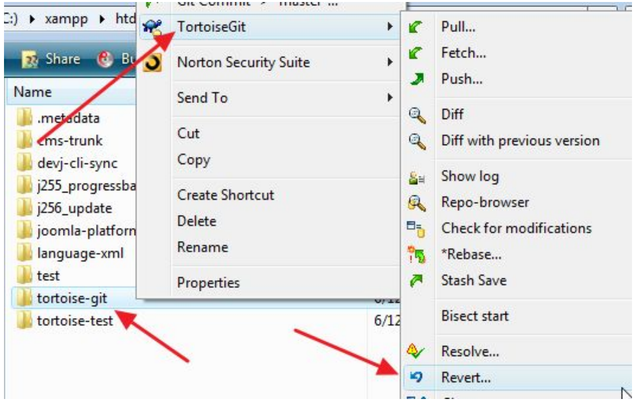


- <u>Cannot</u> be undone!

Undo local changes: revert

To undo the changes you haven’t added and committed yet:

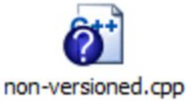
- Right click on the file
- Select TortoiseGit -> Revert
- Select the file that was modified and needs to be restored



- <u>Cannot</u> be undone!
- Works to undo “Add” as well: added file will be unstaged

Undo local renaming

If you rename a file, it will show up as an unstaged file with question mark icon ·



- Right click on a PARENT folder
- Select TortoiseGit -> Revert
- Select a missing file you would like to restore
- You can delete a renamed (unstaged) version of a file now

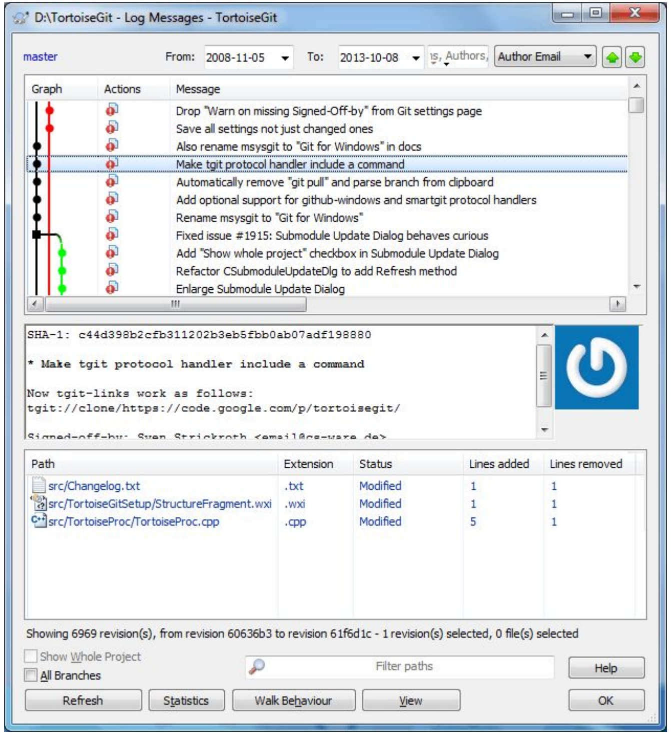
Undo local deletion

If you deleted a file, you can restore it by reverting the parent folder, following the same steps as for undoing renaming.

- Right click on a PARENT folder
- Select TortoiseGit -> Revert
- Select a missing file you would like to restore

Show log: select TortoiseGit -> Show Log

You can see all your commits and commit messages!



Graph	Actions	Message
		Drop "Warn on missing Signed-Off-by" from Git settings page
		Save all settings not just changed ones
		Also rename msysgit to "Git for Windows" in docs
		Make tgit protocol handler include a command
		Automatically remove "git pull" and parse branch from clipboard
		Add optional support for github-windows and smartgit protocol handlers
		Rename msysgit to "Git for Windows"
		Fixed issue #1915: Submodule Update Dialog behaves curious
		Add "Show whole project" checkbox in Submodule Update Dialog
		Refactor CSubmoduleUpdateDlg to add Refresh method
		Enlarge Submodule Update Dialog

```
SHA-1: c44d398b2cfb311202b3eb5fbb0ab07adf198880

* Make tgit protocol handler include a command

Now tgit-links work as follows:
tgit://clone/https://code.google.com/p/tortoisegit/

Signed-off-by: Sven Strickroth <email@cs-ware.de>
```

Path	Extension	Status	Lines added	Lines removed
src/Changelog.txt	.txt	Modified	1	1
src/TortoiseGitSetup/StructureFragment.wxi	.wxi	Modified	1	1
src/TortoiseProc/TortoiseProc.cpp	.cpp	Modified	5	1

Showing 6969 revision(s), from revision 60636b3 to revision 61f6d1c - 1 revision(s) selected, 0 file(s) selected

Show log: description

- The top pane shows a list of revisions where changes to the file/folder have been committed. This summary includes the date and time, the person who committed the revision and the start of the log message.

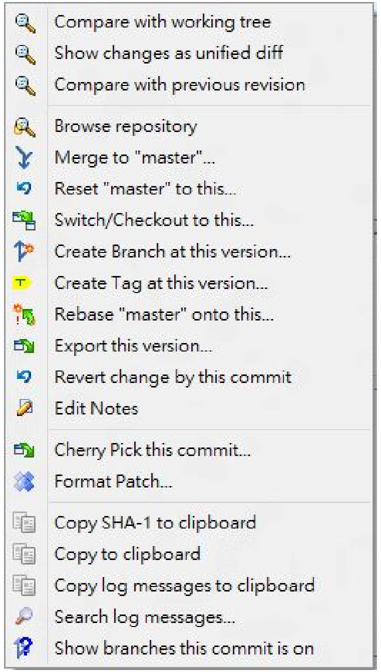
- Lines shown in bold indicate that that is HEAD (for now, current) commit.

- The middle pane shows the full log message for the selected revision.

- The bottom pane shows a list of all files and folders that were changed as part of the selected revision.

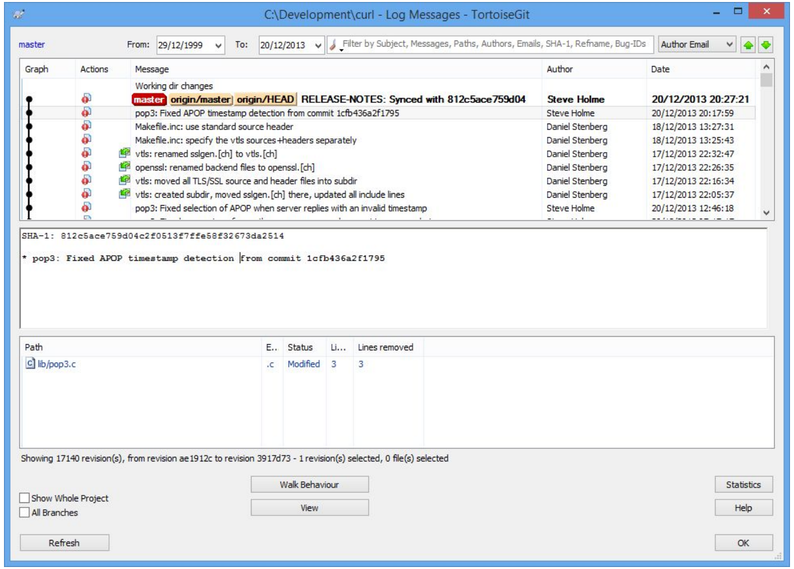
Show log: options

On the top pane, you can right click on any commit and see a new context menu:


Compare with working tree
Show changes as unified diff
Compare with previous revision
Browse repository
Merge to "master"...
Reset "master" to this...
Switch/Checkout to this...
Create Branch at this version...
Create Tag at this version...
Rebase "master" onto this...
Export this version...
Revert change by this commit
Edit Notes
Cherry Pick this commit...
Format Patch...
Copy SHA-1 to clipboard
Copy to clipboard
Copy log messages to clipboard
Search log messages...
Show branches this commit is on

Show log: undo last commit (local repo)

To undo your last commit and delete it from the repo, right click on your last commit (usually called HEAD, in **bold** font)



C:\Development\curl - Log Messages - TortoiseGit
master
From: 29/12/1999
To: 20/12/2013
Filter by Subject, Messages, Paths, Authors, Emails, SHA-1, Refname, Bug-IDs
Author Email

Graph	Actions	Message	Author	Date
		Working dir changes		
		master origin/master origin/HEAD **RELEASE-NOTES: Synced with 812c5ace759d04**	**Steve Holme**	**20/12/2013 20:27:21**
		pop3: Fixed APOP timestamp detection from commit 1cfb436a2f1795	Steve Holme	20/12/2013 20:17:59
		Makefile.inc: use standard source header	Daniel Stenberg	18/12/2013 13:27:31
		Makefile.inc: specify the vtls sources+headers separately	Daniel Stenberg	18/12/2013 13:25:43
		vtls: renamed sslgen.[ch] to vtls.[ch]	Daniel Stenberg	17/12/2013 22:32:47
		openssl: renamed backend files to openssl.[ch]	Daniel Stenberg	17/12/2013 22:26:35
		vtls: moved all TLS/SSL source and header files into subdir	Daniel Stenberg	17/12/2013 22:16:34
		vtls: created subdir, moved sslgen.[ch] there, updated all include lines	Daniel Stenberg	17/12/2013 22:05:37
		pop3: Fixed selection of APOP when server replies with an invalid timestamp	Steve Holme	20/12/2013 12:46:18

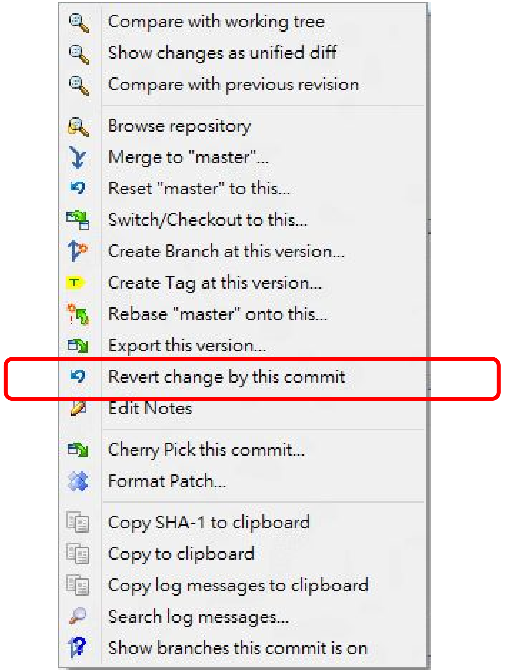
```
SHA-1: 812c5ace759d04c2f0513f7ffe58f32673da2514

* pop3: Fixed APOP timestamp detection from commit 1cfb436a2f1795
```

Path	E..	Status	Li...	Lines removed
lib/pop3.c	.c	Modified	3	3

Showing 17140 revision(s), from revision ae1912c to revision 3917d73 - 1 revision(s) selected, 0 file(s) selected

Show log: undo last commit (local repo)

… and choose “Revert change by this commit”.


Compare with working tree
Show changes as unified diff
Compare with previous revision
Browse repository
Merge to "master"...
Reset "master" to this...
Switch/Checkout to this...
Create Branch at this version...
Create Tag at this version...
Rebase "master" onto this...
Export this version...
Revert change by this commit
Edit Notes
Cherry Pick this commit...
Format Patch...
Copy SHA-1 to clipboard
Copy to clipboard
Copy log messages to clipboard
Search log messages...
Show branches this commit is on

Show log: undo last commit (local repo)

Please, use “Revert changes by this commit” only on the LAST (latest) commit!

Otherwise, you will have errors, because if you choose any old commit before the current commit, you will create a conflict/s:

your newer commits might depend on that old commit!!!

Show log: undo last commit (local repo)

Also, you can’t use it if you have local changes!

Similar reason: your current local changes might depend on the last commit.

Show log: undo last commit (local repo)

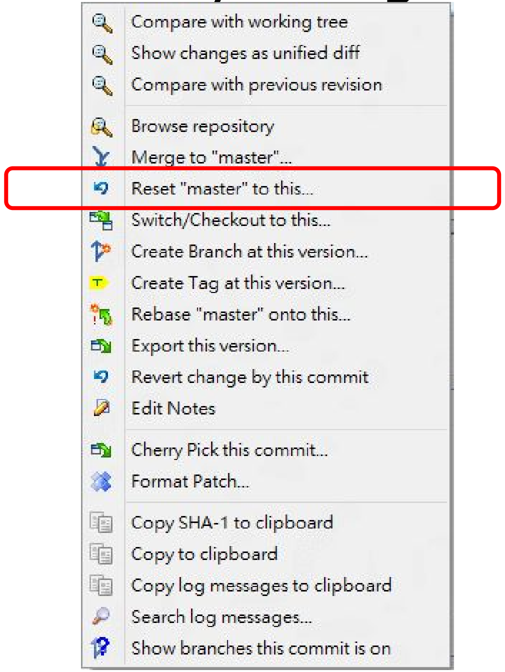
After you chose “Revert changes by this commit”, your repo will look like its files were modified because:

- all the changes made in the last commit were undone
- undoing a change is also a change (opposite action)
- those changes also modify files

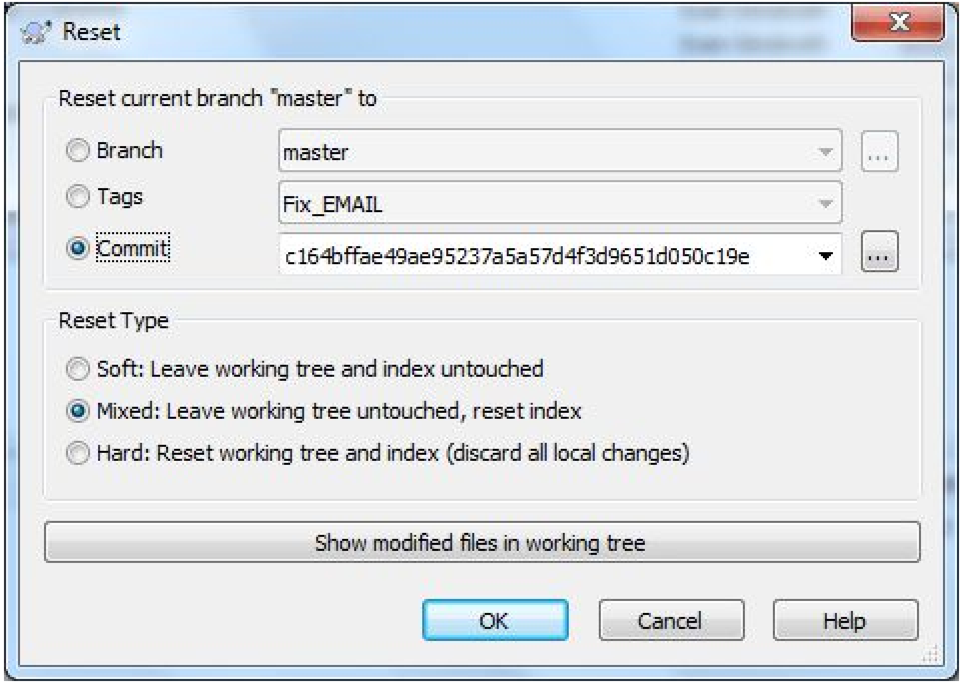
So for the revert to be effective, you **need to commit!**

Show log: undo multiple commits (local repo)

Instead of undoing the latest commit only, you can undo many commits by using “Reset “master” to this”:


Compare with working tree
Show changes as unified diff
Compare with previous revision
Browse repository
Merge to "master"...
Reset "master" to this...
Switch/Checkout to this...
Create Branch at this version...
Create Tag at this version...
Rebase "master" onto this...
Export this version...
Revert change by this commit
Edit Notes
Cherry Pick this commit...
Format Patch...
Copy SHA-1 to clipboard
Copy to clipboard
Copy log messages to clipboard
Search log messages...
Show branches this commit is on

Show log: reset master to this



Reset

Reset current branch "master" to

○ Branch master

○ Tags Fix_EMAIL

● Commit c164bffae49ae95237a5a57d4f3d9651d050c19e

Reset Type

○ Soft: Leave working tree and index untouched

● Mixed: Leave working tree untouched, reset index

○ Hard: Reset working tree and index (discard all local changes)

Show log: reset options

- Soft and mixed: leave local changes (uncommitted changes) untouched
- Hard: discards all local changes

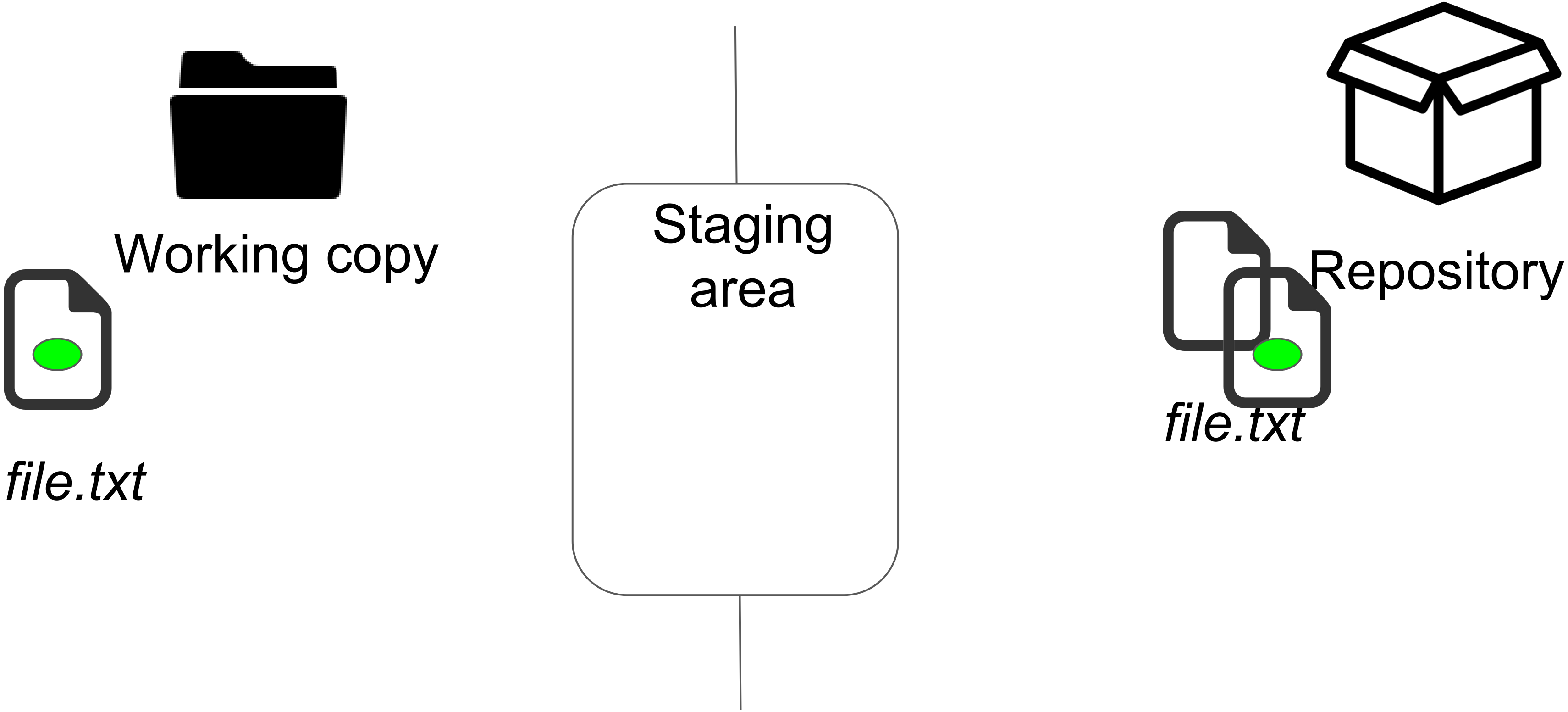
Mixed is a default option!

Remote repo with github.com

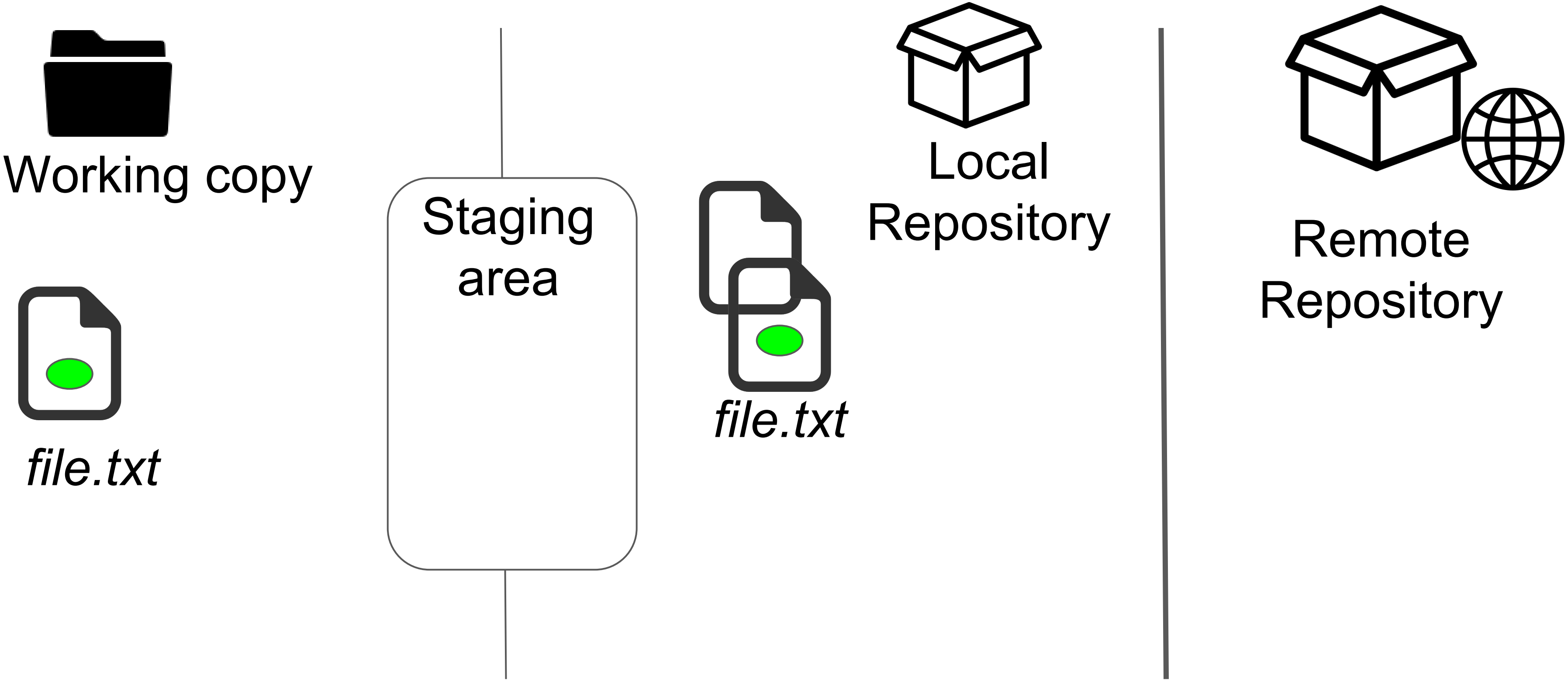
Does everyone have an account for github.com?

Does everyone remember their username and password?

What we have so far:



Remote repo



Working copy

file.txt

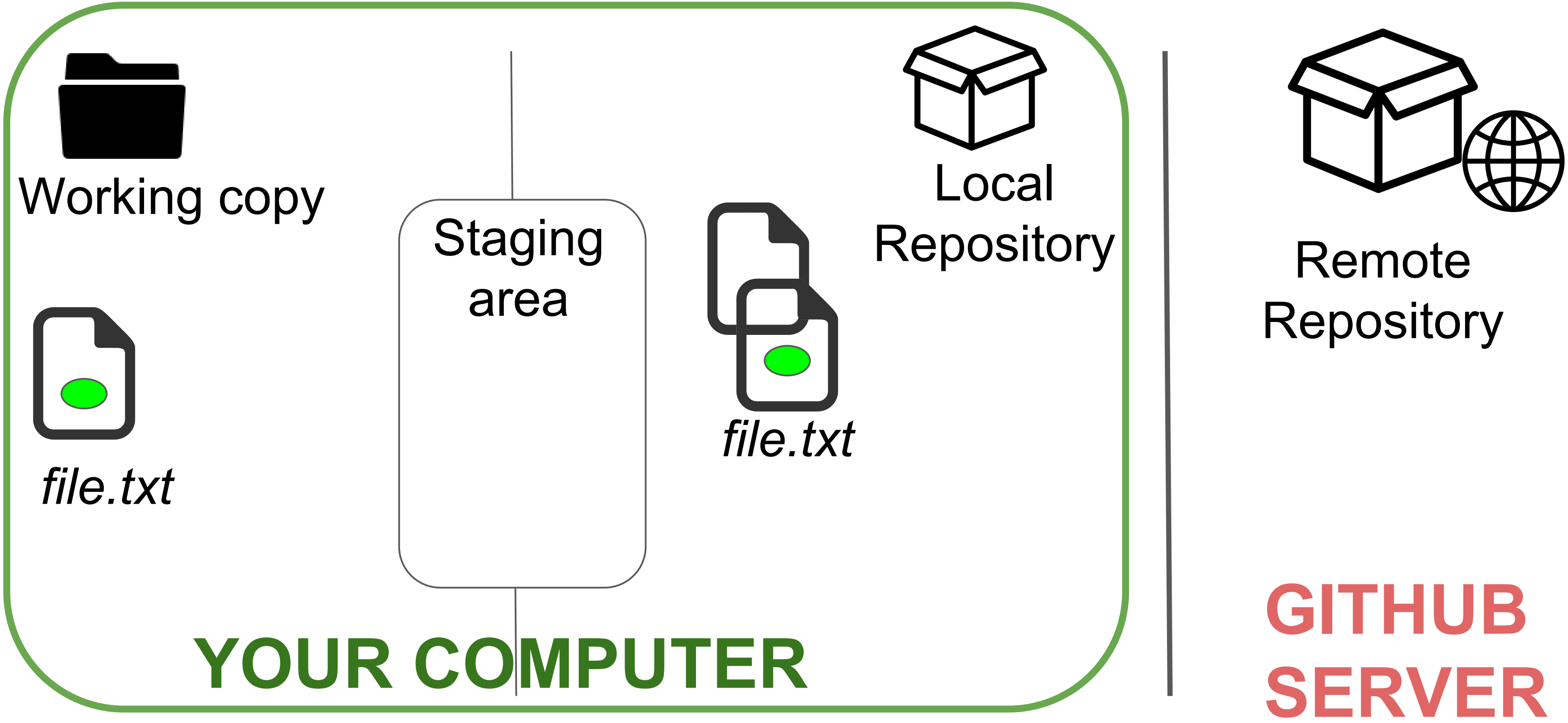
Staging area

file.txt

Local Repository

Remote Repository

Remote repo



Working copy

file.txt

Staging area

file.txt

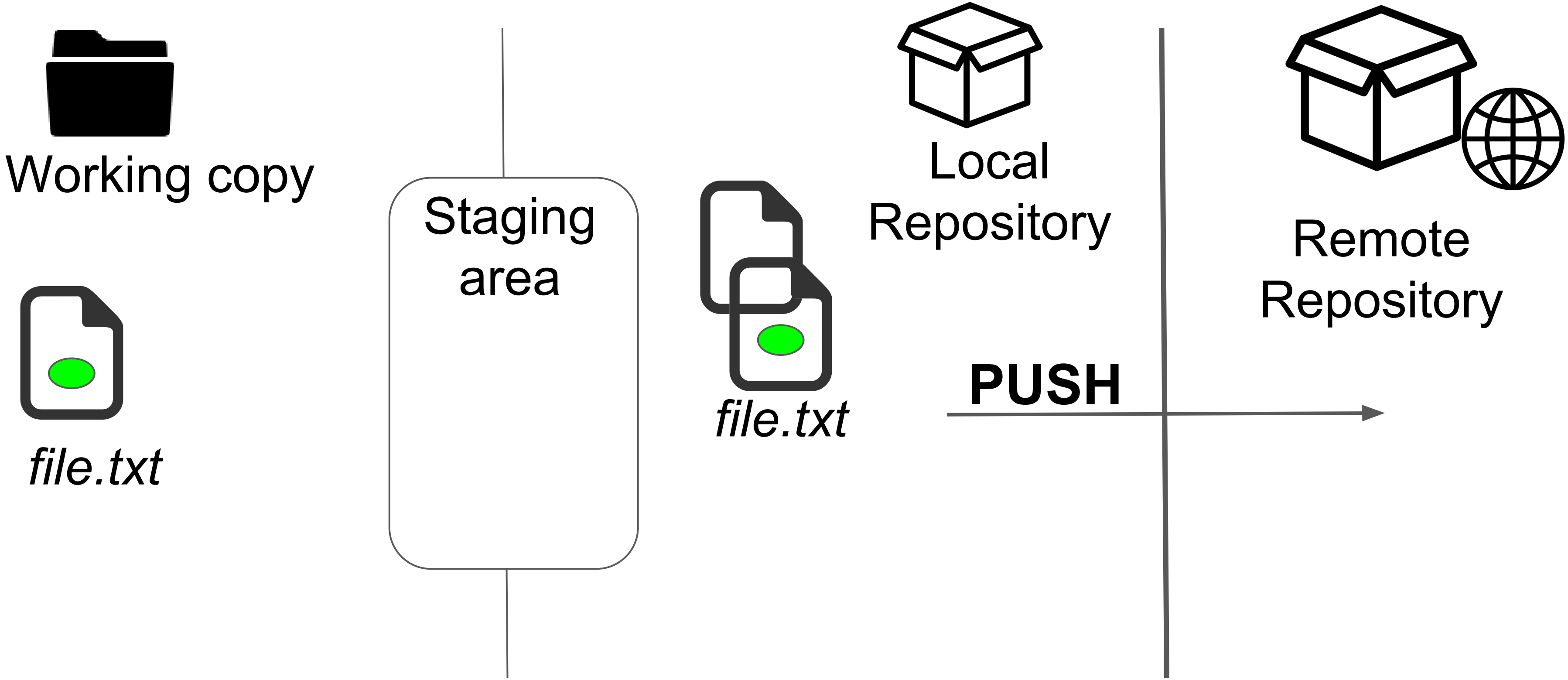
Local Repository

YOUR COMPUTER

Remote Repository

GITHUB SERVER

Remote repo: push changes

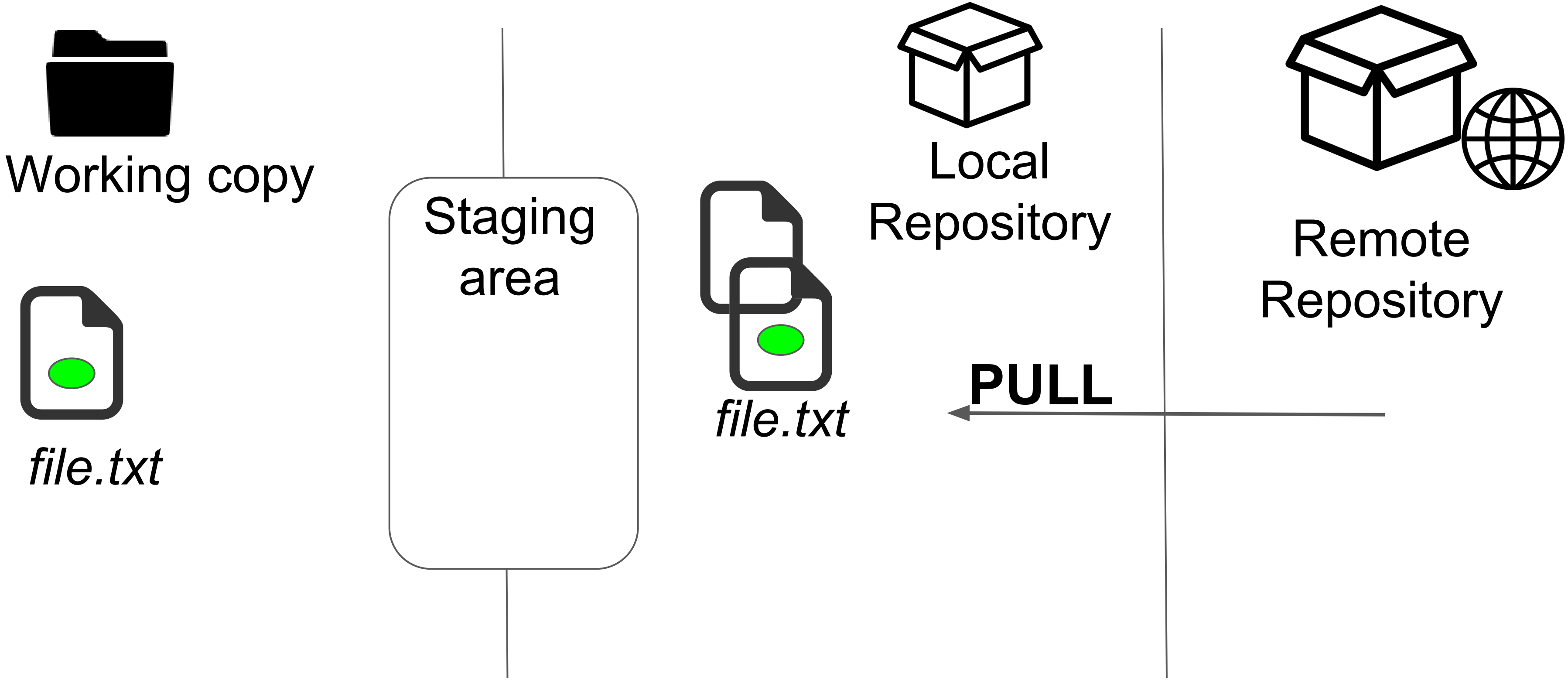


Remote repo: push changes

PUSH:

make all your commits available for other remote repo users.

Remote repo: pull changes

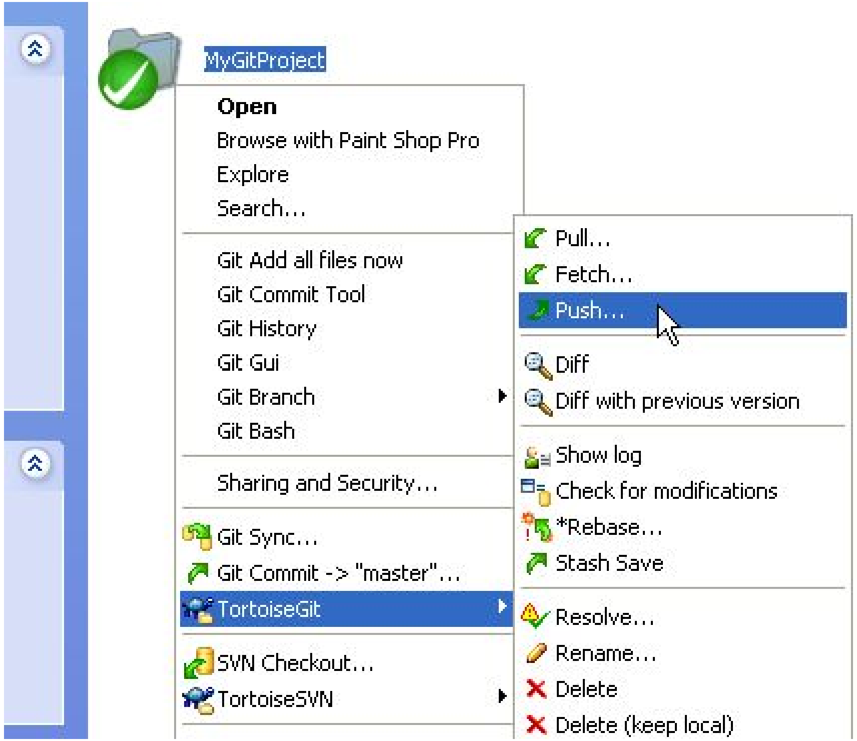


Remote repo: pull changes

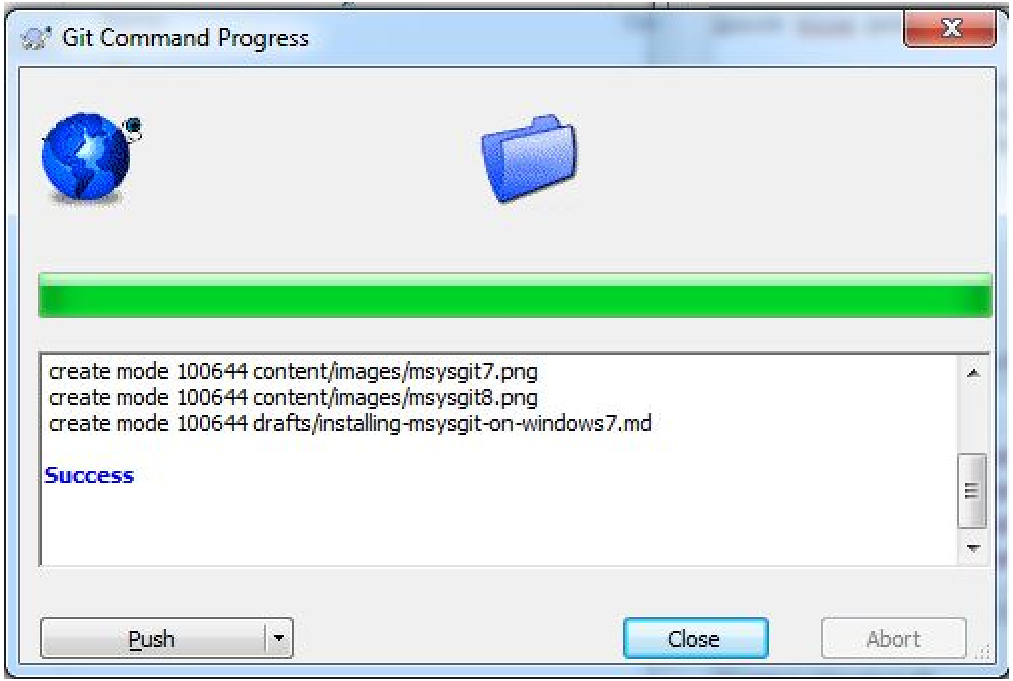
PULL:

upload all the commits other remote repo users pushed to your local repo.

TortoiseGit push and pull

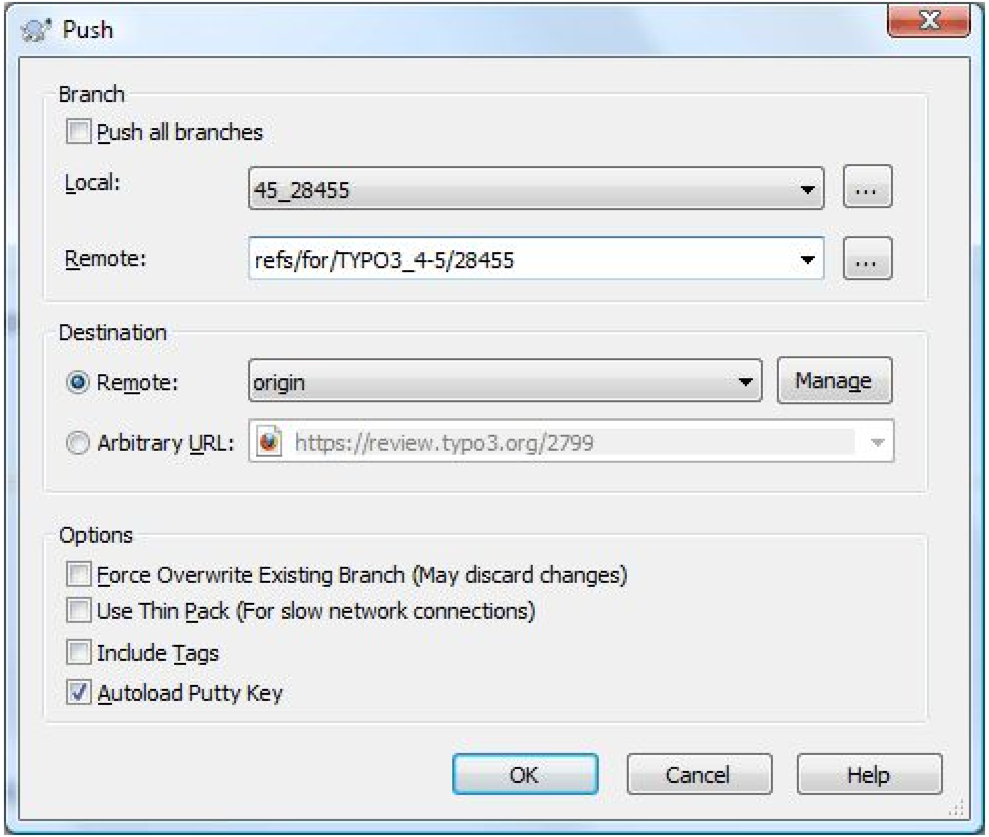


TortoiseGit push and pull



```
create mode 100644 content/images/msysgit7.png
create mode 100644 content/images/msysgit8.png
create mode 100644 drafts/installing-msysgit-on-windows7.md

Success
```

TortoiseGit push and pull



Branch

□ Push all branches

Local: 45_28455

Remote: refs/for/TYPO3_4-5/28455

Destination

● Remote: origin

○ Arbitrary URL: https://review.typo3.org/2799

Options

□ Force Overwrite Existing Branch (May discard changes)

□ Use Thin Pack (For slow network connections)

□ Include Tags

☑ Autoload Putty Key

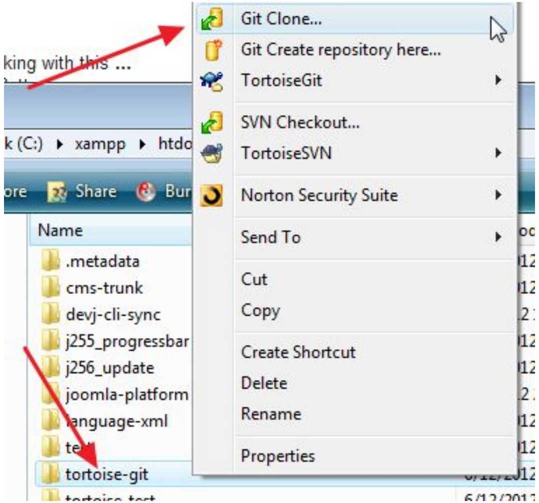
What is remote and what is origin?

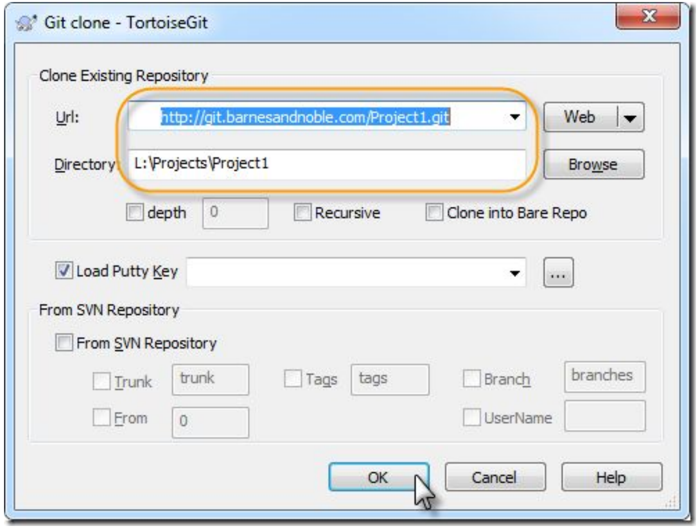
Remote - URL to another copy of your repository.

When you copy another repository, **git** automatically creates a remote named "**origin**" and points to it. **origin** is not the remote repository name. It is an alias given to the URL pointing to the default remote repository

Cloning repo

Clone Repository - make a local copy of some remote repository





Url: http://git.barnesandnoble.com/Project1.git

Directory: L:\Projects\Project1

□ depth 0 □ Recursive □ Clone into Bare Repo

☑ Load Putty Key

□ From SVN Repository

□ Trunk trunk □ Tags tags □ Branch branches

□ From 0 □ UserName

Cloning repo

You can only push to the remote repo if you are granted permissions!

So, before starting exercise 4, come over and give me your username, so I can add you as a collaborator.

Exercise 1: undo local changes

1. Create a new repository, name it “undo_lessons”
2. Add and commit a new file *f1.txt*
3. Write something in *f1.txt* (Do not add/commit your changes yet)
4. Use git to undo your change

5. Rename your file to *f2.txt*
6. Use “Revert” to undo renaming (and delete *f2.txt* after)

7. Delete *f1.txt*
8. Use “Revert” to undo deletion

9. Create a file *f3.txt*, stage (“Add”) it
10. Use “Revert” to undo “Add” action

Exercise 2: undo last commit

1. Delete your *f3.txt* file (we’ll work with *f1.txt* only)

2. Add some more changes to *f1.txt*
3. Commit your changes

4. Using “Show Log” option, choose the latest (last) commit and choose “Revert changes by this commit”

5. Commit your reverting changes

What happened with *f1.txt*? What changes were kept and what changes disappeared?

Exercise 3: undo multiple commits

1. Add some text in *f1.txt*
2. Commit your changes

3. Repeat steps 1 and 2 two times. Now we have many commits.

4. Using “Show Log”, choose one of the oldest commits and choose “Reset “master” to this” (try soft, mixed and hard options!)

How does your file look like now? What happened with recent changes?

Exercise 4: Working with the remote

1. Create a new folder, name it “remote_repo”

2. Use “Git Clone” option and specify the URL: https://github.com/dmukusheva/GTL5CI.git

3. In your copy of the repo, create a new file, name is with **your** name: e.g., *dana*.txt. Write something inside.
4. Add, commit and push your changes.
5. Once everybody is finished, pull the changes.