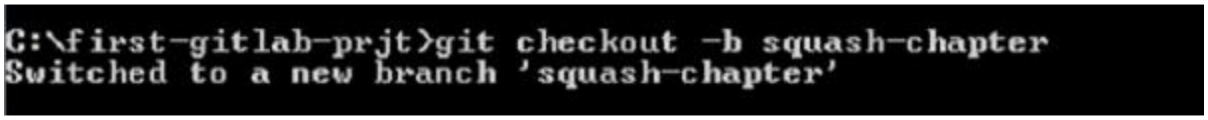
Description

Squashing is a way of combining all commits into one when you are obtaining a merge request.

Steps for Squashing Commits

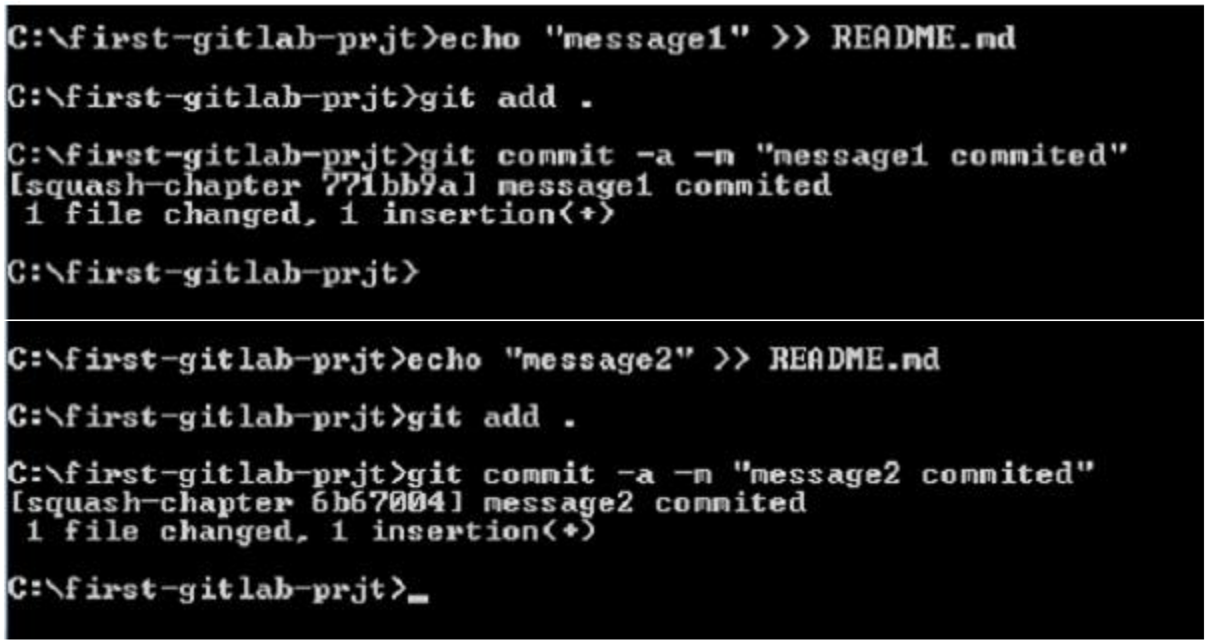
Step 1 − Go to your project directory and check out a new branch with the name *squash-chapter* by using the *git checkout* command −



```
C:\first-gitlab-prjt>git checkout -b squash-chapter
Switched to a new branch 'squash-chapter'
```

The flag *-b* indicates new branch name.

Step 2 − Now, create a new file with two commits, add that file to working directory and store the changes to the repository along with the commit messages as shown below −



```
C:\first-gitlab-prjt>echo "message1" >> README.md

C:\first-gitlab-prjt>git add .

C:\first-gitlab-prjt>git commit -a -m "message1 commited"
[squash-chapter 771bb9a] message1 commited
 1 file changed, 1 insertion(+)

C:\first-gitlab-prjt>
```

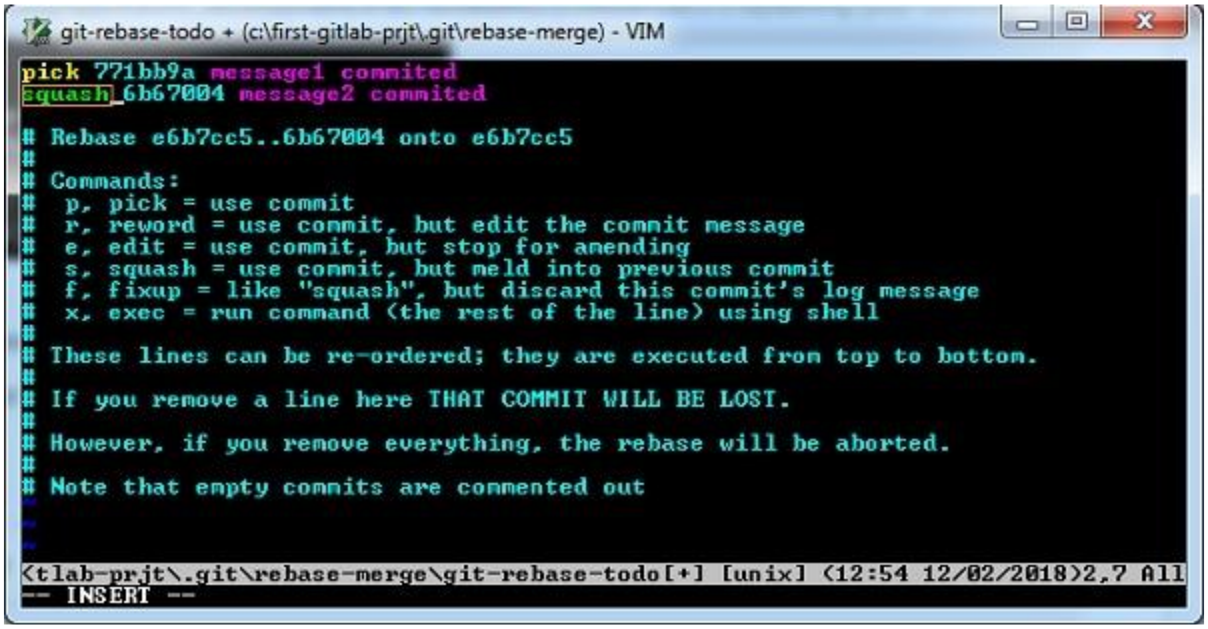
```
C:\first-gitlab-prjt>echo "message2" >> README.md

C:\first-gitlab-prjt>git add .

C:\first-gitlab-prjt>git commit -a -m "message2 commited"
[squash-chapter 6b67004] message2 commited
 1 file changed, 1 insertion(+)

C:\first-gitlab-prjt>_
```

Step 3 − Now, squash the above two commits into one commit by using the below command −

```
$ git rebase -i HEAD~2
```

Here, *git rebase* command is used to integrate changes from one branch to another and *HEAD~2* specifies last two squashed commits and if you want to squash four commits, then you need to write as *HEAD~4*. One more important point is, you need atleast two commits to complete the squash operation.

Step 4 − After entering the above command, it will open the below editor in which you have to change the *pick* word to *squash* word in the second line (you need to squash this commit).

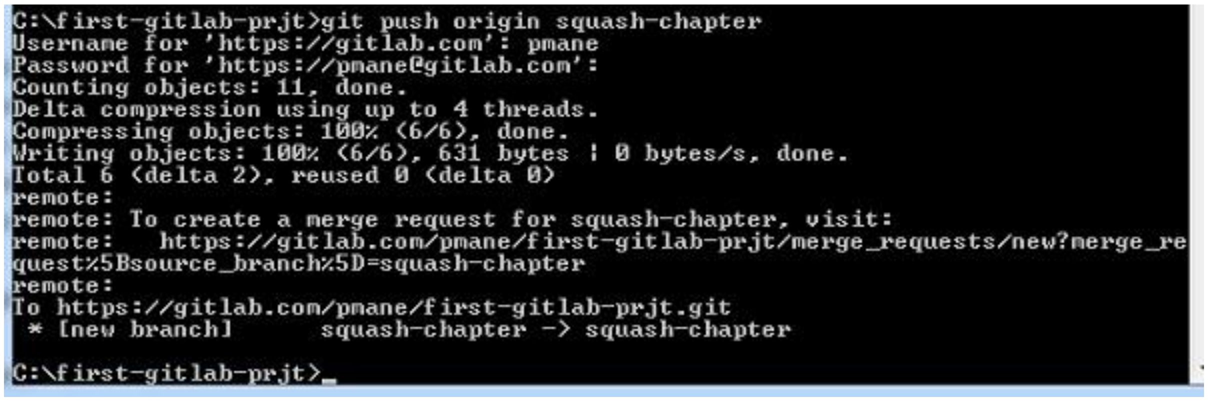


git-rebase-todo + (c:\first-gitlab-prjt\.git\rebase-merge) - VIM

```
pick 771bb9a message1 commited
squash 6b67004 message2 commited

# Rebase e6b7cc5..6b67004 onto e6b7cc5
#
# Commands:
#  p, pick = use commit
#  r, reword = use commit, but edit the commit message
#  e, edit = use commit, but stop for amending
#  s, squash = use commit, but meld into previous commit
#  f, fixup = like "squash", but discard this commit's log message
#  x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
~
~
<tlab-prjt\.git\rebase-merge\git-rebase-todo[+] [unix] (12:54 12/02/2018)2,7 All
-- INSERT --
```

Now press the *Esc* key, then colon(:) and type *wq* to save and exit from the screen.

Step 5 − Now push the branch to remote repository as shown below −



```
C:\first-gitlab-prjt>git push origin squash-chapter
Username for 'https://gitlab.com': pmane
Password for 'https://pmane@gitlab.com':
Counting objects: 11, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 631 bytes | 0 bytes/s, done.
Total 6 (delta 2), reused 0 (delta 0)
remote:
remote: To create a merge request for squash-chapter, visit:
remote:   https://gitlab.com/pmane/first-gitlab-prjt/merge_requests/new?merge_re
quest%5Bsource_branch%5D=squash-chapter
remote:
To https://gitlab.com/pmane/first-gitlab-prjt.git
 * [new branch]      squash-chapter -> squash-chapter

C:\first-gitlab-prjt>_
```