

How can I delete all of my Git stashes at once?

Asked 7 years, 4 months ago Active 2 months ago Viewed 250k times

▲ How can I delete all of my [Git](#) stashes at once?

1266 Specifically I mean, with typing in one command.



★
136

edited Oct 29 '13 at 20:56



[Peter Mortensen](#)

14.6k 19 89 118

asked Jul 6 '12 at 20:38



[Rebekah Waterbury](#)

7,541 5 19 25

5 Answers

▲ The following command deletes all your stashes:

2232 `git stash clear`

▼ From the [git documentation](#):



`clear`

Remove all the stashed states.

IMPORTANT WARNING: Those states will then be subject to pruning, and may be impossible to recover (...).

edited Aug 9 at 2:19



[BKSpurgeon](#)

16.6k 6 64 51

answered Jul 6 '12 at 20:41



[Tadeck](#)

99k 20 129 182

11 why not git stash drop ? – [user20358](#) Sep 2 '15 at 11:48

- 120 @user20358: Because `stash drop` ("*Remove a single stashed state from the stash list. When no <stash> is given, it removes the latest one. (...)*") does not answer the question ("*How can I delete **all** of my Git stashes at once?*")? – [Tadeck](#) Sep 2 '15 at 15:48
- 7 In addition to what accepted answer mentioned - `git stash clear`, to confirm if all cleared, use - `git stash list` – [kamal](#) Sep 20 '17 at 8:15 
- 25 Beware: `stash clear` does not ask you to confirm. – [bishop](#) Mar 30 '18 at 14:55
- 7 @Ryan: `git help <command>` is the best way to get documentation (and, it's actually pretty good). In this case, `git help stash` clearly show that `git stash clear` does the deed (this didn't stop me from coming to SO first and upvoting the answer... but that tells you more about me than `git :D`) – [Marco](#) May 31 '18 at 21:24



24



There are two ways to delete a stash:

1. If you no longer need a particular stash, you can delete it with: `$ git stash drop <stash_id>`.
2. You can delete all of your stashes from the repo with: `$ git stash clear`.

Use both of them with caution, it maybe is difficult to revert the once deleted stashes.

Here is the [reference article](#).

edited Jan 9 at 18:59

answered May 25 '18 at 7:47



[Nesha Zoric](#)

2,534 21 27

- 13 Above answer + comments already cover exactly what your's does... – [Brandon Ling](#) May 25 '18 at 18:28
- 2 Closing the terminal has nothing to do with it. git isn't a daemon; it only runs when you call it. It has no idea you closed the terminal. – [DylanYoung](#) Jan 9 at 13:28



17



this command enables you to look all stashed changes.

```
git stash list
```

Here is the following command use it to clear all of your stashed Changes

```
git stash clear
```

Now if you want to delete one of the stashed changes from stash area

```
git stash drop stash@{index} // here index will be shown after getting stash list.
```

Note : `git stash list` enables you to get index from stash area of git.

edited Mar 27 at 13:05



[ibenjelloun](#)

4,240 1 10 35

answered Feb 27 at 8:40



[Vikrant Kashyap](#)

4,019 1 17 41

I wanted to keep a few recent stashes, but delete everything else.

4

Because all stashes get renumbered when you drop one, this is actually easy to do with while. To delete all stashes older than `stash@{19}`:

```
while git stash drop 'stash@{20}'; do true; done
```

answered Aug 21 at 13:40



[warp](#)

995 8 17

Does this apply to indexed values introduced in Git 2.11, where you can refer to the 3rd stash from the stash list using `git stash pop 3` instead? I wanted to know if we can use the `while` loop and drop stashes using index values. – [tom_mai78101](#) Oct 14 at 14:35

I had another requirement like only few stash have to be removed, below code would be helpful in that case.

3

```
#!/bin/sh
for i in `seq 5 8`
do
    git stash drop stash@{i}
done
```

```
/* will delete from 5 to 8 index*/
```

answered Mar 29 at 12:14

[Bala Krishnan](#)

137 5

```
try for iterator in `seq 5 8`; git stash drop stash@{$iterator}; done
```

 – [Chris McCowan](#) Apr 22 at 20:15 

Note that when you drop, the remaining stashes move up 1 spot. So if you want to remove from 5 to 8, you need to either count backwards or remove `stash@{5}` four times. For example: `for i in `seq 4`; do git stash drop 'stash@{5}'; done` – [warp](#) Aug 21 at 13:36 
