


Is there a quick Git command to see an old version of a file?


▲ Is there a command in Git to see (either dumped to stdout, or in `$PAGER` or `$EDITOR`) a particular version of a particular file?

1366 git

▼

★ 292

edited Oct 19 '18 at 18:56  **Peter Mortensen** 14.2k 19 88 115

asked Dec 3 '08 at 19:31  **mike** 17.4k 40 87 109

5 [How to get a copy of an older version of a file in a git repository?](#) – phuclv Apr 19 '17 at 3:29

9 Answers

▲ You can use `git show` :

1552 `$ git show REVISION:path/to/file`

▼

Replace REVISION with your actual revision (could be a Git commit SHA, a tag name, a branch name, a relative commit name, or any other way of identifying a commit in Git)

For example, to view the version of file `src/main.c` from 4 commits ago, use:

`$ git show HEAD~4:src/main.c`

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



gws

 107 12

mipadi

301k 71 469 453

- 5 That doesn't actually seem to work -- did you try it? For "git show HEAD:path/to/file.c", I get an "ambiguous argument" error. – [mike](#) Dec 3 '08 at 20:06
- 33 Yeah, I tried it out -- it worked for me. – [mipadi](#) Dec 3 '08 at 22:13
- 16 If you're on windows, it might be a path separator thing; if I do git show HEAD:dir\subdir\file, I get the ambiguous argument. If I do git show HEAD:dir/subdir/file, it works as expected. – [Matt McMinn](#) Jul 21 '10 at 14:56
- 10 The path you must provide after the : is from the root of the git repository. (this was given below as an answer but I think it was intended as a comment on this answer) – [MatrixFrog](#) Feb 28 '11 at 19:21
- 9 If you want to see it in a Vim split so that they scroll together, I wrote a short [blog.post](#) showing how to do that. – [Flaviu](#) Dec 25 '11 at 6:18

Doing this by date looks like this:

238

```
git show HEAD@{2013-02-25}:/fileInCurrentDirectory.txt
```

Note that HEAD@{2013-02-25} means "where HEAD was on 2013-02-25" in this repository (using [the reflog](#)), not "the last commit before 2013-02-25 in this branch in history".

edited Apr 26 '15 at 19:20



Matthieu Moy


7,676 1 27 46

answered Mar 7 '13 at 17:05



Jim Hunziker

7,565 6 41 52

- 3 Awesome. Saves my time lot instead of going to github and look at the commit. – [Fizer Khan](#) Sep 15 '14 at 7:45
- The "dot" on the file path was my issue. Thanks! – [tomascharad](#) Jan 19 '15 at 13:15 
- 4 This command is useful with master instead of HEAD@{2013-02-25} , if you're on a branch – [funroll](#) Nov 9 '15 at 13:56
- 1 Can you include the time, à la git log --since='2016-04-28 23:59:59 +0100' ? – [dumbledad](#) May 3 '16 at 12:55

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)

[Sign up with Google](#)
[Sign up with Facebook](#)




108

If you like GUIs, you can use gitk:

1. start gitk with:

```
gitk /path/to/file
```

2. Choose the revision in the top part of the screen, e.g. by description or date. By default, the lower part of the screen shows the diff for that revision, (corresponding to the "patch" radio button).

3. To see the file for the selected revision:

- Click on the "tree" radio button. This will show the root of the file tree at that revision.
- Drill down to your file.

edited Dec 8 '16 at 18:44



Chris Martin

24.5k 4 54 113

answered Dec 19 '12 at 18:21



Trausti Kristjansson

2,209 1 11 11

7 This also works with [tig](#), which is a curses git repo viewer. – [Matthew G](#) May 13 '13 at 5:01

1 @Paul Slocum: May be because this command is not a conventional command, not the built-in of git. I think this command only work for Windows. – [Envil](#) Dec 4 '13 at 2:46

gitk works fine on mac (seems to be Tcl/Tk based). – [Trausti Kristjansson](#) Dec 10 '13 at 3:13

Note this only seems to work if you start from the root of your git repository. – [Marc](#) Mar 23 '16 at 16:43

If you want to check against a certain revision with gitk you could also use this shortcut: `gitk REVISION /path/to/file`. This can come in handy when you want to check against a certain version for instance. – [Christian.D](#) Jul 6 '16 at 7:42



75

You can also specify a `commit hash` (often also called `commit ID`) with the [git show command](#).

In a nutshell

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



1. Show the log of all the changes for a given file with `git log ./path/to/file`
2. In the list of changes shown, it shows the `commit hash` such as `commit 06c98...` (06c98... being the commit hash)
3. Copy the `commit hash`
4. Run the command `git show <commitHash>:/path/to/file` using the `commit hash` of step 3 & the `path/to/file` of step 1.

Note: adding the `./` when specifying a relative path seems important, i.e. `git show b2f8be577166577c59b55e11cfff1404baf63a84:./flight-simulation/src/main/components/nav-horiz.html` .

edited Nov 16 '16 at 16:19

answered Nov 3 '16 at 11:21



Adrien Be

13.5k 12 74 123

1 in case you don't know path to file, use `git show <SHA1> --name-only` to get it. – Tiina Oct 11 '17 at 1:03

this command op - even auto completes from memory - tested on a deleted directory... can't get more op than that gg – treyBake Nov 10 '17 at 12:55

2 The `./` was necessary for me; thank you. – Anomaly Feb 16 '18 at 13:40



In addition to [Jim Hunziker](#)'s answer,

37

you can export the file from the revision as,



```
git show HEAD@{2013-02-25}:./fileInCurrentDirectory.txt > old_fileInCurrentDirectory.txt
```

Hope this helps :)

edited May 23 '17 at 12:26



Community ♦

1 1

answered Apr 30 '14 at 5:07



Ijas Ameenudeen

7,147 3 31 48

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

Sign up with Google

Sign up with Facebook



edited May 6 '16 at 20:09

answered May 6 '16 at 20:02



sanbor

952 12 20

3 Looks like `git log -pm` would also show merge commits. – [sanbor](#) May 6 '16 at 20:07

You can also run `git log -p -- filename.txt` to restrain the history to only the desired file. – [Jean Paul](#) Jan 21 at 12:25

To quickly see the differences with older revisions of a file:

8

`git show -1 filename.txt >` to compare against the last revision of file

`git show -2 filename.txt >` to compare against the 2nd last revision

`git show -3 filename.txt >` to compare against the last 3rd last revision

edited May 28 at 6:26

answered Dec 13 '18 at 6:05



sachin_ur

524 5 17

7 Those commands show the differences with the current version for me but not show the entire file. – [Jean Paul](#) Jan 21 at 11:31

You can use a script like this to dump all the versions of a file to separate files:

3

e.g.


```
git_dump_all_versions_of_a_file.sh path/to/somefile.txt
```

Get the [script here](#) as an answer to another similar question

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)

1 git_root , git_log_short and git_log_message_for_commit are missing. – [mogsie](#) Jan 26 '18 at 14:23

Good catch! I double posted this answer to 2 different spots, and just removed this one and linked to the other one, where people told me about this before... thanks @mogsie ! – [Brad Parks](#) Jan 26 '18 at 14:32 

This script is very useful! – [XO](#) Dec 22 '18 at 3:33

Helper to fetch multiple files from a given revision

0

When trying to resolve merge conflicts, this helper is very useful:

```
#!/usr/bin/env python3
```

```
import argparse
import os
import subprocess
```

```
parser = argparse.ArgumentParser()
parser.add_argument('revision')
parser.add_argument('files', nargs='+')
args = parser.parse_args()
toplevel = subprocess.check_output(['git', 'rev-parse', '--show-
toplevel']).rstrip().decode()
for path in args.files:
    file_relative = os.path.relpath(os.path.abspath(path), topLevel)
    base, ext = os.path.splitext(path)
    new_path = base + '.old' + ext
    with open(new_path, 'w') as f:
        subprocess.call(['git', 'show', '{}:./{}'.format(args.revision, path)],
            stdout=f)
```

[GitHub upstream](#).

Usage:

```
git-show-save other-branch file1.c path/to/file2.cpp
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



This way, you keep the file extension so your editor won't complain, and can easily find the old file just next to the newer one.

edited Dec 7 '18 at 18:55

answered Dec 7 '18 at 10:54



Ciro Santilli 新疆改造中心996ICU六四事件

161k 36 612 505

you created a python app to basically run a command line. – [Mickey Perlstein](#) Mar 24 at 14:48

@MickeyPerlstein if you can make achieve the same interface with a better implementation, I'm all ears. – [Ciro Santilli](#) 新疆改造中心996ICU六四事件 Mar 24 at 16:38

maybe i don't understand (and if so, my apologies) but isn't it just : "git show version:./path > new_path " ? – [Mickey Perlstein](#) Mar 26 at 11:34

@MickeyPerlstein hi, yes, my command generates that CLI, but it loops over multiple files and produces output name from input, so you don't have to type too much. Nothing revolutionary of course, but convenient. – [Ciro Santilli](#) 新疆改造中心996ICU六四事件 Mar 26 at 21:16 ✎

protected by [cs95](#) Dec 20 '18 at 4:51

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus](#) does not count).

Would you like to answer one of these [unanswered questions](#) instead?

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook

