

We're committed to [working with you to build the future of Stack Overflow](#). Your input matters in our "Through the loop" survey.

## Find size of git repo

Asked 8 years ago   Active 1 month ago   Viewed 114k times



265



git

filesize



64

What's a simple way to find the size of my git repository? And I don't mean `du -h` on the root directory of my repo. I have a lot of ignored files so that size would be different from my total repo size. I essentially want to know how much data would be transferred upon cloning my repo.

edited Nov 23 '15 at 8:19



BH2017

961 2 11 28

asked Nov 18 '11 at 16:03



mschallert

2,919 2 13 10

[See the size of a github repo before cloning it ?](#) – azzamsa Mar 6 '18 at 21:48

### 7 Answers



246



**UPDATE** git 1.8.3 introduced a more efficient way to get a rough size: `git count-objects -vH` (see answer by @VonC)

For different ideas of "complete size" you could use:

```
git bundle create tmp.bundle --all
du -sh tmp.bundle
```



Close (but not exact:)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

With the latter, you would also be counting:

- hooks
- config (remotes, push branches, settings (whitespace, merge, aliases, user details etc.)
- stashes (see [Can I fetch a stash from a remote repo into a local branch?](#) also)
- rerere cache (which can get considerable)
- reflogs
- backups (from filter-branch, e.g.) and various other things (intermediate state from rebase, bisect etc.)

edited May 23 '17 at 12:26



Community ♦

1 1

answered Nov 18 '11 at 16:06



sehe

291k

36

370

493

Thanks! The first behavior more accurately reflected the total clone size but the second was also pretty close. – [mschallert](#) Nov 18 '11 at 16:14

7 Note: this answer is out of date. See VonC's [answer](#) instead for git versions >= 1.8.3. – [John Dibling](#) Nov 14 '14 at 21:43 ✎

Note also my answer is a ways more complete, which could have it's own merit. – [sehe](#) May 17 '16 at 12:47

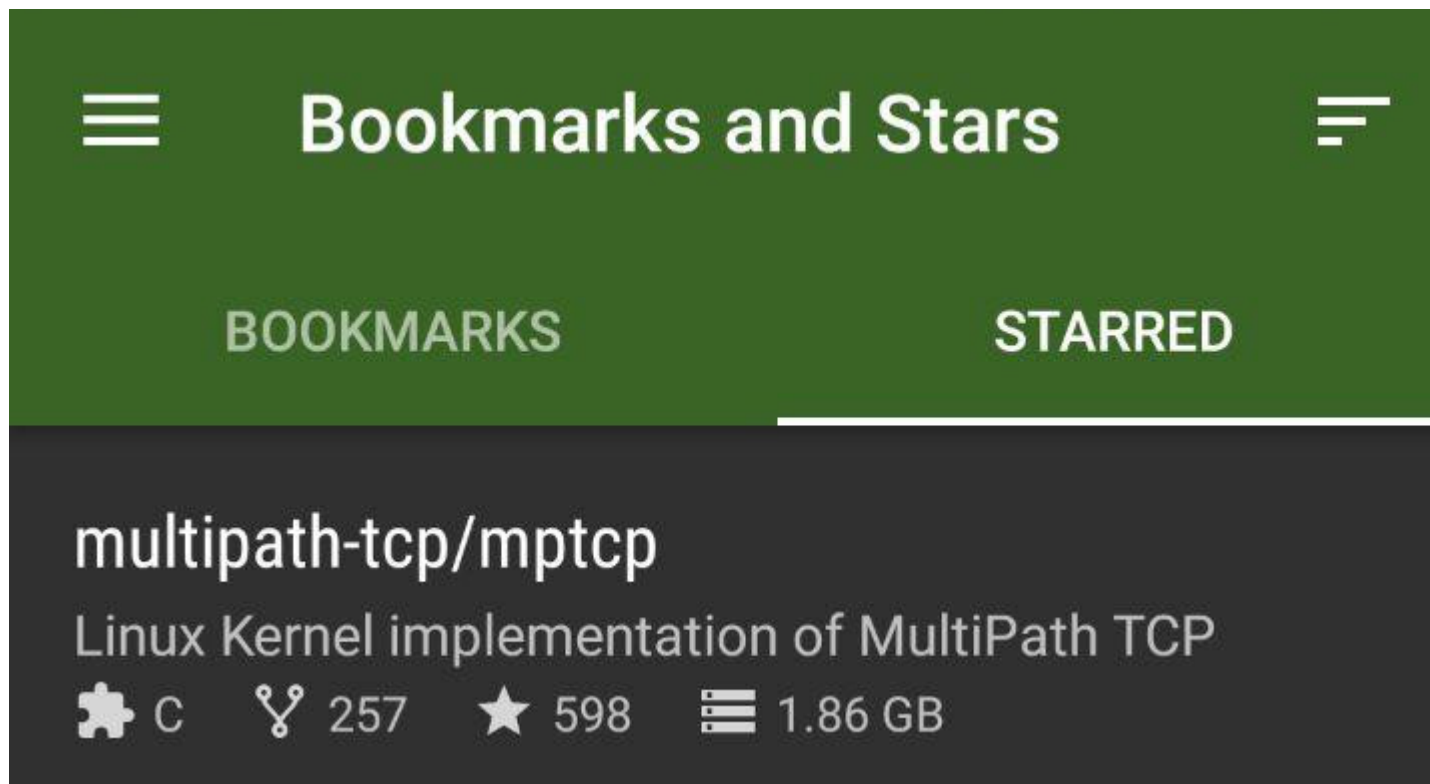
Can you please add what `git bundle` and `git gc` actually does? will it create another (unwanted) archive and then measure the size? – [Ramesh Pareek](#) May 25 '16 at 11:02

1 @RameshPareek yes. This is pretty clear because I'm using `du` to measure the size of that file. – [sehe](#) May 25 '16 at 11:48



If the repository is on [github](#), you can use the open source android app [octodroid](#) which displays the approximate size of the repository by default. For example with the [mptcp](#) repository

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



```
git clone git@github.com:RMPR/mptcp.git
Cloning into 'mptcp'...
remote: Enumerating objects: 39, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (26/26), done.
Receiving objects: 98% (6827909/6925734), 1.71 GiB |
```

Disclaimer : I didn't create octodroid.

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



I *think* this gives you the total list of all files in the repo history:

1 

```
git rev-list --objects --all | git cat-file --batch-check="%(<objectsize>) %(<rest>)" | cut -d" " -f1 | paste -s -d + - | bc
```

You can replace `--all` with a treeish ( `HEAD` , `origin/master` , etc.) to calculate the size of a branch.

answered Jun 1 at 3:06



Mehrdad

174k 96 435 778

You could use [git-sizer](#) . In the `--verbose` setting, the example output is (below). Look for the `Total size of files` line.

1

```
$ git-sizer --verbose
Processing blobs: 1652370
Processing trees: 3396199
Processing commits: 722647
Matching commits to trees: 722647
Processing annotated tags: 534
Processing references: 539
```

Name	Value	Level of concern
Overall repository size		
* Commits		
* Count	723 k	*
* Total size	525 MiB	**
* Trees		
* Count	3.40 M	**
* Total size	9.00 GiB	****
* Total tree entries	264 M	*****
* Blobs		
* Count	1.65 M	*
* Total size	55.8 GiB	*****
* Annotated tags		
* Count	534	

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
| Biggest objects
| * Commits
|   * Maximum size      [1] | 72.7 KiB | *
|   * Maximum parents   [2] | 66       | *****
| * Trees
|   * Maximum entries   [3] | 1.68 k   | *
| * Blobs
|   * Maximum size      [4] | 13.5 MiB | *
|
| History structure
| * Maximum history depth
| * Maximum tag depth   [5] | 1        |
|
| Biggest checkouts
| * Number of directories [6] | 4.38 k   | **
| * Maximum path depth   [7] | 13       | *
| * Maximum path length   [8] | 134 B    | *
| * Number of files       [9] | 62.3 k   | *
| * Total size of files   [9] | 747 MiB  |
| * Number of symlinks    [10] | 40       |
| * Number of submodules  | 0        |
|
[1] 91cc53b0c78596a73fa708cceb7313e7168bb146
[2] 2cde51fbd0f310c8a2c5f977e665c0ac3945b46d
[3] 4f86eed5893207aca2c2da86b35b38f2e1ec1fc8 (refs/heads/master:arch/arm/boot/dts)
[4] a02b6794337286bc12c907c33d5d75537c240bd0
(refs/heads/master:drivers/gpu/drm/amd/include/asic_reg/vega10/NBIO/nbio_6_1_sh_mask.h)
[5] 5dc01c595e6c6ec9ccda4f6f69c131c0dd945f8c (refs/tags/v2.6.11)
[6] 1459754b9d9acc2ffac8525bed6691e15913c6e2
(589b754df3f37ca0a1f96fccde7f91c59266f38a^{tree})
[7] 78a269635e76ed927e17d7883f2d90313570fdb
(dae09011115133666e47c35673c0564b0a702db7^{tree})
[8] ce5f2e31d3bdc1186041fdfd27a5ac96e728f2c5 (refs/heads/master^{tree})
[9] 532bdadc08402b7a72a4b45a2e02e5c710b7d626
(e9ef1fe312b533592e39cddc1327463c30b0ed8d^{tree})
[10] f29a5ea76884ac37e1197bef1941f62fda3f7b99
(f5308d1b83eba20e69df5e0926ba7257c8dd9074^{tree})
```

answered Sep 19 '18 at 4:01

[serv-inc](#)

20.8k 6 99 102

1 This is nice because it shows more information than just the total cloning size. One thing it doesn't show that I'd still like to know is how big the ZIP will

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

@IndrajeetGour: you need to install from [github.com/github/git-sizer/#getting-started](https://github.com/github/git-sizer/#getting-started). Downloads are at [github.com/github/git-sizer/releases](https://github.com/github/git-sizer/releases). – serv-inc Jan 11 at 13:06

Would the @downvoter care to explain his reasoning, or not? – serv-inc Feb 5 at 3:37



1



If you use git LFS, git count-objects does not count your binaries, but only the pointers to them.

If your LFS files are managed by Artifactorys, you should use the REST API:

- Get the [www.jfrog.com](http://www.jfrog.com) API from any search engine
- Look at Get Storage Summary Info

edited Oct 6 '17 at 11:56

answered Oct 5 '17 at 10:04



elike

31 3



233



Note that, since [git 1.8.3 \(April, 22d 2013\)](#):

" git count-objects " learned " --human-readable " aka " -H " option to show various large numbers in Ki / Mi / GiB scaled as necessary.

That could be combined with the -v option mentioned by [Jack Morrison](#) in [his answer](#).

```
git gc
git count-objects -vH
```

( [git\\_gc](#) is important, as mentioned by [A-B-B's answer](#) )

Plus (still git 1.8.3), the output is more complete:

" git count-objects -v " learned to **report leftover temporary packfiles and other garbage in the object store.**

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



1 Worked like a charm thanks @VonC – [niri9428480](#) Nov 13 at 6:32



## The git command

107

```
git count-objects -v
```



will give you a good estimate of the git repository's size. Without the `-v` flag, it only tells you the size of your unpacked files. This command may not be in your `$PATH`, you may have to track it down (on Ubuntu I found it in `/usr/lib/git-core/`, for instance).

From the Git man-page:

`-v, --verbose`

In addition to the number of loose objects and disk space consumed, it reports the number of in-pack objects, number of packs, disk space consumed by those packs, and number of objects that can be removed by running `git prune-packed`.

Your output will look similar to the following:

```
count: 1910
size: 19764
in-pack: 41814
packs: 3
size-pack: 1066963
prune-packable: 1
garbage: 0
```

The line you're looking for is `size-pack`. That is the size of all the packed commit objects, or the smallest possible size for the new cloned repository.

edited Dec 19 '12 at 13:44




[user1338062](#)

answered Nov 18 '11 at 16:23



[Jack Morrison](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

- 
- 4 In my experience, `git count-objects -v` works when `git-count-objects -v` does not (because it is not in my PATH). – [mwolfetech](#) Aug 7 '12 at 16:55 
- 
- It doesn't work as expected, produces always size-pack: 0 – [psihodelia](#) Mar 28 '13 at 16:37
- 
- 5 @psihodelia It seems you might have to run `git gc` before running `git count-objects -v` because your repository may not have been packed yet. See the first answer here for evidence: [stackoverflow.com/questions/3532740/...](https://stackoverflow.com/questions/3532740/...) – [Jack Morrison](#) Mar 29 '13 at 20:20
- 
- 5 You can use `grep` to get only the desired line: `git count-objects -vH | grep 'size-pack'`. The `H` is to show it in human readable format, as @VonC said in his answer. – [alko989](#) Jun 18 '14 at 11:44
- 
- 2 size-pack in KB ? – [Steven Du](#) Apr 26 '15 at 7:12
-