How to list all the files in a commit?



I am looking for a simple git command that provides a nicely formatted list of all files that were part of the commit given by a hash (SHA1), with no extraneous information.

2498

I have tried:



git show a303aa90779efdd2f6b9d90693e2cbbbe4613c1d



Although it lists the files, it also includes unwanted diff information for each.

Is there another git command that will provide just the list I want, so that I can avoid parsing it from the git show output?





asked Jan 8 '09 at 12:26



- I came here looking for something a bit different. I want to see all files modified for a set of commits and wound up using git log --until 2013-05-21 --pretty="short" --name-only with a good effect. Limited Atonement Dec 12 '13 at 17:40
- 3 Use this command to get all changes from previous n commits till master: git diff-tree --name-status -r @{3} master ako Jun 22 '18 at 13:57

git diff --name-only master - To list ALL changed files on current branch, comparing to master branch. - Noam Manos Jun 30 at 9:39

29 Answers

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

Sign up with email





javascript/application.js
javascript/ie6.js

Another Way (less preferred for scripts, because it's a porcelain command; meant to be user-facing)

```
$ git show --pretty="" --name-only bd61ad98
index.html
javascript/application.js
javascript/ie6.js
```

- The --no-commit-id suppresses the commit ID output.
- The --pretty argument specifies an empty format string to avoid the cruft at the beginning.
- The --name-only argument shows only the file names that were affected (Thanks Hank).
- The -r argument is to recurse into sub-trees

edited Jun 7 '16 at 19:18

answered Jan 8 '09 at 13:02



Ryan McGeary 198k 10 84 9

- 20 It should be noted that diff-tree won't work when looking at the root commit. jbranchaud Mar 6 '13 at 5:52
- 297 Replacing the --name-only option with --name-status will give more clear summary. Kurt Zhong Apr 11 '13 at 3:58
- 17 If you want it to work on the root commit, use the --root flag. From the man page: "When --root is specified the initial commit will be shown as a big creation event. This is equivalent to a diff against the NULL tree." Chris Sep 23 '13 at 18:26
- git log --name-only -n 1 <hash> The last commit would be: git log --name-only -n 1 HEAD~1..HEAD Kurt Nov 11 '13 at 4:20 /
- If anyone is wondering (like I was) why the first way is "preferred," it goes back to @drizzt 's comment; git show is "porcelain" (meant to be user facing) and git diff-tree is "plumbing" (meant to be used programmatically, e.g. from scripts). The interface for the former may change over time (so the git maintainers could drop --name-only although I don't imagine they would) for useability reasons, whereas the interface for the latter will be kept as stable as possible for compatibility reasons. killscreen Jun 29 '15 at 4:42

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

Sign up with email





If you want to get list of all files in a commit, you can use

git ls-tree --name-only -r <commit-ish>

edited Jan 18 '13 at 12:48

answered Jan 14 '09 at 19:22



Jakub Narębski 224k 55 199

- 1 The Is-tree with --name-only does not seem to work on 1.6.4.4 or 1.6.3.3. Do you think this is a bug? krosenvold Oct 10 '09 at 10:20
- 2 It turns out the ordering of the parameters is significant here. The one in your post does not work, while the one in your response *does* work at least until you update your post;) krosenvold Oct 10 '09 at 15:17
- 5 Pass --no-commit-id to avoid printing the SHA1, like so: git diff-tree --no-commit-id --name-only -r <commit-ish> John Mellor Aug 15 '12 at 20:00
- 3 @CoDEmanX : You didn't miss adding -r / -t option, did you? Because diff-tree handles both modified and added files. If you want to list all new (added) files, use git diff-tree -r --name-only --no-commit-id --diff-filter=A <commit-ish> − Jakub Narębski Nov 17 '15 at 23:15 ✓
- 1 Shouldn't this be trivial? Niklas Rosencrantz Jun 23 '16 at 14:43



I'll just assume that gitk is not desired for this. In that case, try git show --name-only <sha> .

201



answered Jan 8 '09 at 12:34



Hank Gay

53.5k 25 138 213

- 28 --name-only is plenty in most cases where i needed it; Therefore, upvoted the shortest solution (and the only one that i'd remember in 1 try). Erik S Aug 1 '12 at 21:42
- 21 Or --name-status . Neil Traft Mar 18 '15 at 1:15

As someone who really likes CLI git, gitk is actually a decent way of reviewing the files and displaying the file that the diff is on. e.g. Code reviewing a monster commit from a peer. – Elijah Lynn Oct 27 '17 at 21:56

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

Sign up with email









```
git show --stat --oneline HEAD
git show --stat --oneline b24f5fb
git show --stat --oneline HEAD^^..HEAD
```

If you do not like/want the addition/removal stats, you can replace --stat with --name-only

```
git show --name-only --oneline HEAD git show --name-only --oneline b24f5fb git show --name-only --oneline HEAD^^..HEAD
```

edited Jan 24 '18 at 15:15



answered Jul 11 '12 at 23:21



- 3 This is great! It basically gives you the file summary that Github shows at the top of a commit view. Thanks. trisweb Jan 11 '13 at 17:17
 - Very nice. To define an alias: alias gits='git show --stat --oneline', then gits by itself shows the latest changes (in HEAD), while gits b24f5fb can be used to show any revision's changes. Brent Faust Dec 5 '13 at 3:03
- 2 One could also create a git alias... e.g. perhaps git config --global alias.changes 'show --stat --oneline'. Then you can type git changes (with an optional commit-ish) and get the output from the first examples above. lindes Dec 5 '13 at 17:31
 - Git for Windows requires double quotes: git config --global alias.changes "show --stat --oneline" Alchemistmatt Jul 25 '17 at 2:46
- 2 Nice. And unlike the accepted answer, git show also works for reviewing stashed changes: e.g. git show --stat --oneline stash@{1} Jeff Ward Jul 26 '17 at 19:00



You can also do

68 git log --name-only



and you can browse through various commits, commit messages and the changed files.

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

Sign up with email



1 like. Nice n' simple :D − kmiklas Feb 13 at 19:47 🖍



Recently I needed to list all changed files between two commits. So I used this (also *nix specific) command

65

git show --pretty="format:" --name-only START COMMIT..END COMMIT | sort | uniq



Update: Or as Ethan points out below

git diff --name-only START_COMMIT..END_COMMIT

Using --name-status will also include the change (added, modified, deleted etc) next to each file

git diff --name-status START_COMMIT..END_COMMIT

edited Jun 21 '15 at 7:25

answered Apr 26 '10 at 12:23



lunohodov **2,660** 2 21 17

2 Just tried: Works fine in git bash on Windows too. – OregonGhost Aug 16 '12 at 16:52

4 If you use git diff --name-status START COMMIT..END COMMIT then you don't need the trailing | sort | uniq . - Ethan Sep 17 '13 at 20:12

Correction to above comment: git diff --name-only START_COMMIT..END_COMMIT - Ethan Sep 17 '13 at 20:19

This is what I was looking for. How I used it: git diff --name-only START_COMMIT..END_COMMIT | grep -v -e '**.png' -e '**.xml' . I wanted a list of code changes only for a huge PR that had added thousands of PNGs and XML layouts. - AutonomousApps Sep 18 '17 at 16:39



Simplest form:

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

Sign up with email



```
git show --stat --name-only (hash)
```

edited Aug 17 '16 at 4:11

answered Jul 16 '16 at 0:06



--name-only will still include a couple of header lines containing information such as the author, date and the commit message. - devoured elysium Sep 29 '16 at 15:14

only one I found that works for the merge commit – Alex Punnen Jul 8 at 10:15 ▶



I use **changed** alias a quite often. To set it up:



git config --global alias.changed 'show --pretty="format:" --name-only'



then:

```
git changed (lists files modified in last commit)
git changed bAda55 (lists files modified in this commit)
git changed bAda55..ff0021 (lists files modified between those commits)
```

Similar commands that may be useful:

```
git log --name-status --oneline (very similar, but shows what actually happened M/C/D)
git show --name-only
```

edited Apr 30 '13 at 7:31

answered Apr 29 '13 at 13:13



takeshin

♣ ♣ 30.9k 26 107 154

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

Sign up with email





If you want also the status of the changed files:

git diff --name-status <sha>^ <sha>

This works well with merge commits.

edited Mar 10 '17 at 10:39

answered Apr 29 '11 at 14:53





Use

31 git log --name-status



This will show you the commit id, message, the files changed and whether it was modified, created, added or deleted. Somewhat of a all-in-one command.

answered May 15 '18 at 16:42



1 use --oneline . makes it easier to read - Suhaib Feb 5 at 9:56



\$ git log 88ee8^..88ee8 --name-only --pretty="format:"

24



answered Jan 8 '09 at 14:11



Pat Notz

138k 28 82 90

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

Sign up with email



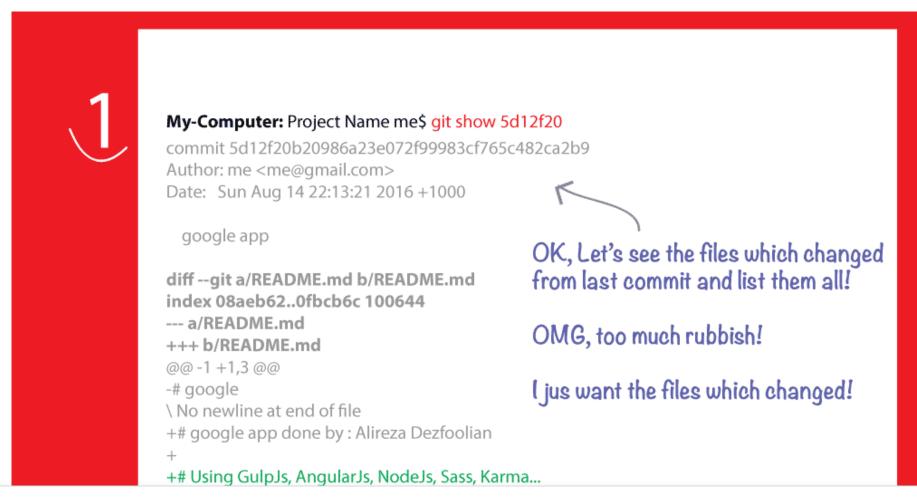
19

To reduce the info and **show only names** of the files which committed, you simply can add --name-only or --name-status flag..., these flags just show you the file names which are different from previous commits as you want...

So you can do git diff followed by --name-only, with two commit hashes after <sha0> <sha1>, something like below:

```
git diff --name-only 5f12f15 kag9f02
```

I also create the below image to show all steps to go through in these situation:



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

Sign up with email





My-Computer: Project Name me\$ git diff 5d12f20 daa9f14

commit 5d12f20b20986a23e072f99983cf765c482ca2b9

Author: me <me@gmail.com>

Date: Sun Aug 14 22:13:21 2016 +1000

google app

diff --git a/README.md b/README.md index 08aeb62..0fbcb6c 100644

--- a/README.md

+++ b/README.md

@@ -1 +1,3 @@

-# google

\ No newline at end of file

+# google app done by : Alireza Dezfoolian

+

+# Using GulpJs, AngularJs, NodeJs, Sass, Karma...

\ No newline at end of file

diff --git //jsx/main.jsx b/jsx/main.jsx

new file mode 100644

2

My-Computer: Project Name me\$ git diff --name-only 5d12f20 daa9f14

css/main.css gulpfile.js index.html js/main.js



Ok, let's use --name-only flag this time!

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

Sign up with email



Sign up with Facebook



Ok, let's try compare 2 latest commits...

Ohhhhhh.... same again!

edited Dec 7 '17 at 2:27

answered Jul 6 '17 at 13:32



Alireza

58.6k 14 194 127



I use this to get list of modified files between two changesets:

git diff --name-status <SHA1> <SHA2> | cut -f2



answered Mar 28 '10 at 7:37



use --name-only and you can skip the cut command - Newtonx Jul 28 '12 at 0:54

Yeah but the status can be quite handy at well (for isntance, you might want to grep to display all files except those that have been deleted with something like git diff --name-status .. | grep ^[^D] | cut -f2 - Pierre-Adrien Buisson Sep 17 '14 at 9:19 /



I like to use



git show --stat <SHA1>^...<SHA2>



answered Nov 3 '10 at 8:12



Michael De Silva

3,685 1 15 24

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

Sign up with email







Hm, nope: info other than the file list is also displayed, which is what the OP specifically doesn't want. – Dan Cornilescu Jan 12 '16 at 15:37



There's also git whatchanged, which is more low level than git log

NAME

git-whatchanged - Show logs with difference each commit introduces

It outputs the commit summary with a list of files beneath it with their modes and if there added(A), deleted(D) or modified(M);

\$ git whatchanged f31a441398fb7834fde24c5b0c2974182a431363

Would give something like:

```
commit f31a441398fb7834fde24c5b0c2974182a431363
Author: xx <xx@xx.nl>
Date: Tue Sep 29 17:23:22 2015 +0200
   added fb skd and XLForm
:000000 100644 0000000... 90a20d7... A Pods/Bolts/Bolts/Common/BFCancellationToken.h
:000000 100644 0000000... b5006d0... A Pods/Bolts/Bolts/Common/BFCancellationToken.m
:000000 100644 0000000... 3e7b711... A
Pods/Bolts/Bolts/Common/BFCancellationTokenRegistration.h
:000000 100644 0000000... 9c8a7ae... A
Pods/Bolts/Bolts/Common/BFCancellationTokenRegistration.m
:000000 100644 0000000... bd6e7a1... A
Pods/Bolts/Bolts/Common/BFCancellationTokenSource.h
:000000 100644 0000000... 947f725... A
Pods/Bolts/Bolts/Common/BFCancellationTokenSource.m
:000000 100644 0000000... cf7dcdf... A Pods/Bolts/Bolts/Common/BFDefines.h
:000000 100644 0000000... 02af9ba... A Pods/Bolts/Bolts/Common/BFExecutor.h
:000000 100644 0000000... 292e27c... A Pods/Bolts/Bolts/Common/BFExecutor.m
:000000 100644 0000000... 827071d... A Pods/Bolts/Bolts/Common/BFTask.h
```

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

Sign up with email





answered Oct 14 '15 at 11:05



16.6k 5 66 72

Plus, just one command whatchanged instead of supplying parameters. - Gabrielius Jan 25 '16 at 12:19



I like this:

git diff --name-status <SHA1> <SHA1>^



edited Aug 12 '09 at 18:17

answered Apr 22 '09 at 16:10



skiphoppy

39.7k 60 157 204

I think this gets the A & D (add and delete) file statuses backwards, because it's showing the diff from the specified commit to the previous commit, instead of the other way around. It should be git diff --name-status <SHA1>^ <SHA1> . - Troy Gizzi Jan 3 '15 at 5:14 /



Use simple one line command, if you just want the list of files changed in the last commit:

git diff HEAD~1 --name-only



edited May 18 '14 at 22:34



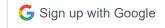
adosaiguas **1.243** 8 13 answered Nov 28 '13 at 15:36



It shows nothing on running this command - Laxmikant Ratnaparkhi Mar 28 '14 at 7:24

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

Sign up with email







This doesn't show log messages, extra newlines, or any other clutter. This works for any commit, not just the current one. Not sure why it hasn't quite been mentioned yet, so I'm adding it.

answered Jul 28 '12 at 0:55



This two looks the same: git diff SHA1^ SHA1 and git show SHA1. - Vladimir Vukanac Jun 27 '16 at 15:16

@mrW Those commands produce similar output, but git show also shows the commit message - Newtonx Jun 27 '16 at 23:01



Display the log.

COMMIT can be blank ("") or the sha-1 or the sha-1 shortened.



git log COMMIT -1 --name-only

This will list just the files, very useful for further processing.

git log COMMIT -1 --name-only --pretty=format:"" | grep "[^\s]"

edited Jun 9 '15 at 10:17



Surya

13.3k 2 35 59

answered Aug 25 '11 at 12:34



thefreshteapot 1 2



Found a perfect answer to this:

git show --name-status --oneline <commit-hash>

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

Sign up with email



Sign up with Facebook

X

Which files were deleted, D

answered Sep 1 '17 at 21:51





A combination of "git show --stat" (thanks Ryan) and a couple of sed commands should trim the data down for you:

5 git show --stat $\langle SHA1 \rangle$ | sed -n "/ [\w]*|/p" | sed "s/|.*\$//"



That will produce just the list of modified files.

edited Jun 9 '15 at 10:17
Surya

13.3k 2 35 59

answered Jan 8 '09 at 13:59





There is a simple trick to view as a file listing, just add: after the hash.

git show 9d3a52c474:



You can then drill in,

git show 9d3a52c474:someDir/someOtherDir

If you hit a file you'll get the raw version of the file; which sometimes is what you want if you're only looking for a nice reference or key pieces of code (diffs can make everything a mess),

git show 9d3a52c474:someDir/someOtherDir/somefile

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

Sign up with email



Sign up with Facebook

X





try this command for name and changes number of line



git show --stat <commit-hash>



only show file names

for get last commit hash then try this command

last commit with show files name and file status modify, create or delete

```
git log -1 --oneline --name-status
```

or for all

git log

edited May 29 at 11:22

answered Oct 6 '18 at 13:20



1 @DanFare "fatal: unrecognized argument: --names-only" from 2.20.1.windows.1 – user2864740 Mar 1 at 18:24

I would correct if I could so instead adding this comment. There is a typo.. It should say: git log -1 --name-only This will get the names of the files

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

Sign up with email







git show HEAD@{0}

works fine for me





13.3k 2 35 59

answered Jun 12 '12 at 9:31



Bruce



If you are using oh-my-zsh and git plugin, the **glg** shortcut is helpful.

edited Jun 1 at 19:05

answered Jan 7 at 0:09



Henry

461 5 15

« ... with **no extraneous** information. » (OP) – RomainValeri May 29 at 12:09 /

Thanks, removed extraneous information. - Henry Jun 1 at 19:05

Did you just... snip all code from your answer? This is not at all what I referred to. No extraneous information in the result output as asked by OP, not in your answer! You still have an answer that outputs A LOT of extraneous information. But now, it's just described a lot less accurately. OP wanted to ouput a list of files, and nothing more. Look at what glg outputs: a lot more. Sorry for the confusion. - RomainValeri Jun 1 at 20:44 /



I use this to get list of changed files in merge commit

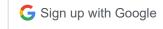
λ git log -m -1 --name-only --pretty="format:" configs/anotherconfig.xml configs/configsInRepo.xml

or

λ git log -m -1 --name-status --pretty="format:"

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

Sign up with email









I thought I would share a summary of my alias.. also I find using 'zsh' great with git it chroma keys everything nicely and tells you want branch are in all of the time by changing the command prompt.

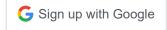
-2

For those covering from SVN you will find this useful: (this is a combination of ideas from different threads, I only take credit of knowing how to use copy/paste)

```
.gitconfig:
        ls = log --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s
%Cgreen(%cr) %C(bold blue)%an%Creset' --abbrev-commit --date=relative --name-status
>>git ls
* 99f21a6 - (HEAD -> swift) New Files from xcode 7 (11 hours ago) Jim Zucker
        icds.xcodeproj/project.pbxproj
       icds.xcodeproj/project.xcworkspace/contents.xcworkspacedata
Α
       icds/AppDelegate.m
Α
       icds/Assets.xcassets/AppIcon.appiconset/Contents.json
A
* e0a1bb6 - Move everything to old (11 hours ago) Jim Zucker
D
       Classes/AppInfoViewControler.h
D
        Classes/AppInfoViewControler.m
D
       Classes/CurveInstrument.h
.gitconfig:
       lt = log --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s
%Cgreen(%cr) %C(bold blue)%an%Creset' --abbrev-commit --date=relative
>>git lt
* 99f21a6 - (HEAD -> swift) New Files from xcode 7 (11 hours ago) Jim Zucker
* e0a1bb6 - Move everything to old (11 hours ago) Jim Zucker
* 778bda6 - Cleanup for new project (11 hours ago) Jim Zucker
* 7373b5e - clean up files from old version (11 hours ago) Jim Zucker
* 14a8d53 - (tag: 1.x, origin/swift, origin/master, master) Initial Commit (16 hours
ago) Jim Zucker
```

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

Sign up with email





commit 99f21a61de832bad7b2bdb74066a08cac3d0bf3c

Author: Jim Zucker <jim@stratengllc.com> Date: Tue Dec 1 22:23:10 2015 -0800

New Files from xcode 7

icds.xcodeproj/project.pbxproj Α

icds.xcodeproj/project.xcworkspace/contents.xcworkspacedata Α

commit e0a1bb6b59ed6a4f9147e894d7f7fe00283fce8d

Author: Jim Zucker <jim@stratengllc.com> Date: Tue Dec 1 22:17:00 2015 -0800

Move everything to old

- D Classes/AppInfoViewControler.h
- Classes/AppInfoViewControler.m
- D Classes/CurveInstrument.h
- Classes/CurveInstrument.m

answered Dec 2 '15 at 17:40



Jim Zucker

5 9

This should work:

git status



This will show what is not staged and what is staged.

answered Mar 8 '18 at 0:53



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

Sign up with email





protected by Vamsi Prabhala Oct 5 '18 at 19:54

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

