revert-branch-rebase

From: Junio C Hamano <gitster@pobox.com> To: git@vger.kernel.org Subject: [HOWTO] Reverting an existing commit Abstract: In this article, JC gives a small real-life example of using ’git revert’ command, and using a temporary branch and tag for safety and easier sanity checking. Date: Mon, 29 Aug 2005 21:39:02 -0700 Content-type: text/asciidoc Message-ID: <7voe7g3uop.fsf@assigned-by-dhcp.cox.net>

How to revert an existing commit ==========

One of the changes I pulled into the ’master’ branch turns out to break building Git with GCC 2.95. While they were well-intentioned portability fixes, keeping things working with gcc-2.95 was also important. Here is what I did to revert the change in the ’master’ branch and to adjust the ’seen’ branch, using core Git tools and barebone Porcelain.

First, prepare a throw-away branch in case I screw things up.

```
$ git checkout -b revert-c99 master
```

Now I am on the ’revert-c99’ branch. Let’s figure out which commit to revert. I happen to know that the top of the ’master’ branch is a merge, and its second parent (i.e. foreign commit I merged from) has the change I would want to undo. Further I happen to know that that merge introduced 5 commits or so:

```
$ git show-branch --more=4 master master^2 | head
* [master] Merge refs/heads/portable from http://www.cs.berkeley....
 ! [master^2] Replace C99 array initializers with code.
--
-  [master] Merge refs/heads/portable from http://www.cs.berkeley....
*+ [master^2] Replace C99 array initializers with code.
*+ [master^2~1] Replace unsetenv() and setenv() with older putenv().
*+ [master^2~2] Include sys/time.h in daemon.c.
*+ [master^2~3] Fix ?: statements.
*+ [master^2~4] Replace zero-length array decls with [].
*  [master~1] tutorial note about git branch
```

The '--more=4’ above means "after we reach the merge base of refs, show until we display four more common commits". That last commit would have been where the "portable" branch was forked from the main git.git repository, so this would show everything on both branches since then. I just limited the output to the first handful using ’head’.

Now I know ’master^2~4’ (pronounce it as "find the second parent of the ’master’, and then go four generations back following the first parent") is the one I would want to revert. Since I also want to say why I am reverting it, the ’-n’ flag is given to ’git revert’. This prevents it from actually making a commit, and instead ’git revert’ leaves the commit log message it wanted to use in ’.msg’ file:

```
$ git revert -n master^2~4
$ cat .msg
Revert "Replace zero-length array decls with []."

This reverts 6c5f9baa3bc0d63e141e0afc23110205379905a4 commit.
$ git diff HEAD ;# to make sure what we are reverting makes sense.
$ make CC=gcc-2.95 clean test ;# make sure it fixed the breakage.
$ make clean test ;# make sure it did not cause other breakage.
```

The reverted change makes sense (from reading the ’diff’ output), does fix the problem (from ’make CC=gcc-2.95’ test), and does not cause new breakage (from the last ’make test’). I’m ready to commit:

```
$ git commit -a -s ;# read .msg into the log,
                   # and explain why I am reverting.
```

I could have screwed up in any of the above steps, but in the worst case I could just have done ’git checkout master’ to start over. Fortunately I did not have to; what I have in the current branch ’revert-c99’ is what I want. So merge that back into ’master’:

```
$ git checkout master
$ git merge revert-c99 ;# this should be a fast-forward
Updating from 10d781b9caa4f71495c7b34963bef137216f86a8 to e3a693c...
 cache.h        |    8 ++++----
 commit.c       |    2 +-
 ls-files.c     |    2 +-
 receive-pack.c |    2 +-
 server-info.c  |    2 +-
 5 files changed, 8 insertions(+), 8 deletions(-)
```

There is no need to redo the test at this point. We fast-forwarded and we know ’master’ matches ’revert-c99’ exactly. In fact:

```
$ git diff master..revert-c99
```

says nothing.

Then we rebase the ’seen’ branch as usual.

```
$ git checkout seen
$ git tag seen-anchor seen
$ git rebase master
* Applying: Redo "revert" using three-way merge machinery.
First trying simple merge strategy to cherry-pick.
* Applying: Remove git-apply-patch-script.
First trying simple merge strategy to cherry-pick.
Simple cherry-pick fails; trying Automatic cherry-pick.
Removing Documentation/git-apply-patch-script.txt
Removing git-apply-patch-script
* Applying: Document "git cherry-pick" and "git revert"
First trying simple merge strategy to cherry-pick.
* Applying: mailinfo and applymbox updates
First trying simple merge strategy to cherry-pick.
* Applying: Show commits in topo order and name all commits.
First trying simple merge strategy to cherry-pick.
* Applying: More documentation updates.
First trying simple merge strategy to cherry-pick.
```

The temporary tag ’seen-anchor’ is me just being careful, in case ’git rebase’ screws up. After this, I can do these for sanity check:

```
$ git diff seen-anchor..seen ;# make sure we got the master fix.
$ make CC=gcc-2.95 clean test ;# make sure it fixed the breakage.
$ make clean test ;# make sure it did not cause other breakage.
```

Everything is in the good order. I do not need the temporary branch or tag anymore, so remove them:

```
$ rm -f .git/refs/tags/seen-anchor
$ git branch -d revert-c99
```

It was an emergency fix, so we might as well merge it into the ’release candidate’ branch, although I expect the next release would be some days off:

```
$ git checkout rc
$ git pull . master
Packing 0 objects
Unpacking 0 objects
```

```
* commit-ish: e3a693c...    refs/heads/master from .
Trying to merge e3a693c... into 8c1f5f0... using 10d781b...
Committed merge 7fb9b7262a1d1e0a47bbfdcbbcf50ce0635d3f8f
 cache.h        |    8 ++++----
 commit.c       |    2 +-
 ls-files.c     |    2 +-
 receive-pack.c |    2 +-
 server-info.c  |    2 +-
 5 files changed, 8 insertions(+), 8 deletions(-)
```

And the final repository status looks like this:

```
$ git show-branch --more=1 master seen rc
! [master] Revert "Replace zero-length array decls with []."
 ! [seen] git-repack: Add option to repack all objects.
  * [rc] Merge refs/heads/master from .
---
 +  [seen] git-repack: Add option to repack all objects.
 +  [seen~1] More documentation updates.
 +  [seen~2] Show commits in topo order and name all commits.
 +  [seen~3] mailinfo and applymbox updates
 +  [seen~4] Document "git cherry-pick" and "git revert"
 +  [seen~5] Remove git-apply-patch-script.
 +  [seen~6] Redo "revert" using three-way merge machinery.
  - [rc] Merge refs/heads/master from .
++* [master] Revert "Replace zero-length array decls with []."
  - [rc~1] Merge refs/heads/master from .
... [master~1] Merge refs/heads/portable from http://www.cs.berkeley....
```

□□□ #2
□ □□□ □□□ 2026-10-31 20:15:07 UTC
□ □□□ □□□ 2026-10-31 20:15:08 UTC