ariejan de vroom

How to create and apply a patch with Git

Monday 26 October 2009

Git is quite common nowadays and a lot of people are asking me how they can create a patch file. Creating a patch file with git is quite easy to do, you just need to see how it's done a few times.

This article will show you how to create a patch from the last few commits in your repository. Next, I'll also show you how you can correctly apply this patch to another repository. ~ **Before you start**

To make creating patches easier, there are some common git practices you should follow. It's not necessary, but it will make your life easier.

If you fix a bug or create a new feature – do it in a separate branch!

Let's say you want to create a patch for my imdb gem. You should clone my repository and create a new branch for the fix you have in mind. In this sample we'll do an imaginary fix for empty posters.

```
git clone git://github.com/ariejan/imdb.git
cd imdb
git checkout -b fix_empty_poster
```

Now, in the new `fix_empty_poster` branch you can hack whatever you need to fix. Write tests, update code etc. etc.

When you're satisfied with all you changes, it's time to create your patch. FYI: I'm assuming you made a few commits in the `fix_empty_poster` branch and did *not* yet merge it back in to the `master` branch.

Creating the patch

Okay, I've made some commits, here's the `git log` for the `fix_empty_poster` branch:

```
git log --pretty=oneline -3
* ce30d1f - (fix_empty_poster) Added poster URL as part of cli output (7 minutes ago)
* 5998b80 - Added specs to test empty poster URL behaviour (12 minutes ago)
* aecb8cb - (REL-0.5.0, origin/master, origin/HEAD, master) Prepare release 0.5.0 (4 months ago)
```

In GitX it would look like this:

Okay, now it's time to go and make a patch! All we really want are the two latest commits, stuff them in a file and send them to someone to apply them. But, since we created a separate branch, we don't have to worry about commits at all!

```
git format-patch master --stdout > fix_empty_poster.patch
```

This will create a new file `fix_empty_poster.patch` with all changes from the current (`fix_empty_poster`) against `master`. Normally, git would create a separate patch file for each commit, but that's not what we want. All we need is a single patch file.

Now, you have a patch for the fix you wrote. Send it to the maintainer of the project ...

Applying the patch

... who will apply the patch you just sent! But, before you do that, there are some other steps you should take.

First, take a look at what changes are in the patch. You can do this easily with `git apply`

```
git apply --stat fix_empty_poster.patch
```

Note that this command does not apply the patch, but only shows you the stats about what it'll do. After peeking into the patch file with your favorite editor, you can see what the actual changes are.

Next, you're interested in how troublesome the patch is going to be. Git allows you to test the patch before you actually apply it.

```
git apply --check fix_empty_poster.patch
```

If you don't get any errors, the patch can be applied cleanly. Otherwise you may see what trouble you'll run into. To apply the patch, I'll use `git am` instead of `git apply`. The reason for this is that `git am` allows you to *sign off* an applied patch. This may be useful for later reference.

```
git am --signoff < fix_empty_poster.patch
Applying: Added specs to test empty poster URL behaviour
Applying: Added poster URL as part of cli output
```

Okay, patches were applied cleanly and you're master branch has been updated. Of course, run your tests again to make sure nothing got borked.

In you git log, you'll find that the commit messages contain a "Signed-off-by" tag. This tag will be read by Github and others to provide useful info about how the commit ended up in the code.

That's all folks!

Are there any other git topics you'd like covered here? Please let me know!

Did you enjoy this post? Your generosity is much appreciated.

Donate Bitcoins

Recent posts on ariejan.net

2014-10-15 Rails: Prevent Accidental Debugging Commits

2014-08-29 Synchronize goroutines in your tests

2014-06-04 GPG Sign Your Git Commits

2014-04-15 Testing $HOME with Cucumber and Aruba

2014-04-04 Dealing With Technical Debt

51 Comments Ariejan.net Login

Sort by Best Share Favorite

Join the discussion…

Marcus • 5 years ago

Best git patch blog post I've seen so far! For once patching is easy to understand by newcomers to Git. Thanks!

26 • Reply • Share ›

Ariejan de Vroom Mod • 5 years ago

As an added bonus, you can combine the `git format-patch` and `git am` commands to select commits from one branch and selectively apply them to the current one.

```
git format-patch -k --stdout R1..R2 | git am -3 -k
```

11 • Reply • Share ›

Julian Picht → Ariejan de Vroom • 2 years ago

just for people stumling onto this page years later:
git cherry-pick [commit]
is what you really want to use now, to selectively add commits from one branch to another.

25 • Reply • Share ›

realtebo → Julian Picht • 7 months ago

can you do an example ?

• Reply • Share ›

NewWorld → realtebo • 4 months ago

If you google "git cherry pick tutorial" you will find plenty of examples.

• Reply • Share ›

Alex Colina • 2 years ago

"who will apply the patch you just sent! "
I will sound like a dumbass but....

what am I supposed to do? download the patch? if so where do i put it?
and how do I refer to it?

or is it more like:
cd path\to\your\local\repo
git checkout youbranch
git patch https://location/of/remote/patch.patch
git apply...etc...etc...etc
?
Thanks

6 • Reply • Share ›

Héctor Andrés Urbina Saavedra • 3 years ago

I think it would be useful to add the instruction

git checkout master

before

git apply --check fix_empty_poster.patchbecause it's not obvious to do that for someone that doesn't know what "checkout" is forand just wants to create a patch file and check that everything is ok :)

6 • Reply • Share ›

Ludo van den Boom • 5 years ago

Thanks Ariejan! This post is a great guide that helped me a lot with streamlining my patch-create-apply process with git. Aweomse!

3 • Reply • Share ›

Alan B • 2 years ago

Very good post, maybe just adding a section for those who did not have the luxury to have branches so cleanly defined (i.e. patch between commits already in existing commit history)?

1 • Reply • Share ›

Vishnu R • 3 years ago

Thanks alot

1 • Reply • Share ›

Adam G. • 3 years ago

Correct your text:

"Okay, patches were applied cleanly and you're master branch has been updated. Of course, run your tests again to make sure nothing got borked."

Note:
"and you're" -> "and your"
"nothing got borked" -> "nothing got broked"

:)
3 • Reply • Share ›

P'yata Pavel ➦ Adam G. • 3 years ago
"Broked?" You gonna be kidding me :)
27 • Reply • Share ›

jr ➦ P'yata Pavel • 2 years ago
ya, I like 'borked' better
3 • Reply • Share ›

Michael Ketchel ➦ jr • 3 months ago
Borked is a classic alternative to "broken" in the programmer community :P
• Reply • Share ›

Kevin ➦ Adam G. • a year ago
Broked isn't a word.

Borked is a word, but it's used incorrectly here.
2 • Reply • Share ›

Samir Pani • 3 years ago
I have Three Repository " Test" "Stage" and "Live" So do i have to create the patch again and again on the repository if yes...plz suggest me.....?
2 • Reply • Share ›

Bruce Perens • 4 years ago
Thanks, this was useful today. - Bruce Perens
2 • Reply • Share ›

Francois Faure • 2 months ago
How to revert the git am if I find out that something got borked and I want to reject the patch ? Thanks !
• Reply • Share ›

pilkjaer ➦ Francois Faure • 2 months ago
I guess the patch created a new commit in your repo, right? In this case you can just revert it. "git revert <sha>"
1 • Reply • Share ›

MKS • 3 months ago
This was awesome, thanks xx
• Reply • Share ›

Murali A Varma • 4 months ago
Great tutorial. Did my first patch today, and signed off. I feel like a baws!
• Reply • Share ›

Greg Pickett • 5 months ago
If that doesn't work, try: `patch -p1 < file.patch`
• Reply • Share ›

Chriss Kępiński • 9 months ago
great tutorial, thanks!
• Reply • Share ›

rakesh • a year ago
Thank u very much. It is really useful.
• Reply • Share ›

Jitendra • a year ago
thanks for a wonderful and quick tutorial
• Reply • Share ›

Mike • a year ago
Good stuff, thanks for posting!
• Reply • Share ›

Viren • a year ago
Many thanks. Very helpful.
• Reply • Share ›

rahul • a year ago
it is very urgent
• Reply • Share ›

rahul • a year ago
how to setup path for git ple... help me any one
• Reply • Share ›

Isaac Moore • a year ago
Thanks! I didn't expect it to be this easy.
• Reply • Share ›

Nitin Rastogi • 2 years ago
I have .diff files instead of .patch files....

How can I apply these patches in .diff files using GIT ?

Thanks in advance...
• Reply • Share ›

Nitin Rastogi • 2 years ago

I have .diff files instead of .patch files.

How can I apply these patches using GIT ?

• Reply • Share ›

Atharva Patel • 2 years ago

It was really helpful to me. Thanks for the post!

• Reply • Share ›

Amir Elouti • 2 years ago

great information, helped alot
thank you

• Reply • Share ›

Bots ka Bap • 2 years ago

And What happens if you update your code in between? Do i need to update my branch with your code? if so , how?

• Reply • Share ›

Tyler • 2 years ago

Straight and to the point -- thanks.

• Reply • Share ›

Mads Kristensen • 2 years ago

Thanks for this nice guide!

• Reply • Share ›

David Trejo • 2 years ago

Some of the images in this post got broken, still useful though :)

• Reply • Share ›

Millisami • 3 years ago

Best patch creation and applification of git patch command.

• Reply • Share ›

antoinecomte • 3 years ago

nice tutorial. Thanks

• Reply • Share ›

magesh • 3 years ago

Nice!

• Reply • Share ›

Scribble Geek ➦ magesh • 3 years ago

:)

• Reply • Share ›

Scribble Geek ➦ magesh • 3 years ago

tanq

• Reply • Share ›

Gerke Forcare • 3 years ago

See http://www.winksaville.com/blo... for when the patching fails and you need to resolve things.

• Reply • Share ›

chaitali shah • 3 years ago

hey, m getting error here.....

$ git am --signoff < patch3.patchPatch does not have a valid e-mail address.

Can some1 pls guide me.

• Reply • Share ›

Praveen Paneri ➦ chaitali shah • 3 years ago

use patch -p1 instead

• Reply • Share ›

chirantan • 4 years ago

Is it possible to create a patch without having to commit the code? For example, I make a few changes and I want to migrate to another machine and take my changes with me in a patch without having to commit them. Is that possible?

• Reply • Share ›

hamada ➦ chirantan • 4 years ago

it is possible, in that case use only :
git diff > result.patch

2 • Reply • Share ›

Paul Noden ➦ hamada • 3 years ago

you will need to git add any new files before running this command.

1 • Reply • Share ›

Toby Osbourn • 4 years ago

Excellent post, really helped me to understand the process.

• Reply • Share ›

Load more comments