

{ on programming :

« [Some tips for getting started on Project Euler](#)
[Are you in the text editor dark ages?](#) »

Writing better commit messages

If you and your team are in the habit of writing good commit messages, it can save everyone a tremendous amount of time and effort. Other developers can very quickly see what you've done. It'll also save tons of time tracking down commits from the past when you need to find them. A good commit message serves exactly two purposes: it gives other developers an explanation of the commit, and it gives some documentation for future searching.

You aren't special

Everyone thinks they write great messages. Most of the time you probably do. But everyone, including you, phones it in from time to time. It's true, admit it. I know you think you write awesome commits but if we looked at your last 100, how many would we truly classify as awesome? Just for fun, try it. I just did. It wasn't pretty.

Make "atomic" commits

This isn't really about the message, per se, but it's too important to not mention. A commit should be one (and just one) logical unit. It should be something that someone might want to patch or revert in its entirety, and never piecewise. If it could be useful in pieces, make separate commits. This will result in short, clear, and concise commit messages. Non-atomic commits make for awful run-on commit messages.

A run-on commit:

Fixed a few bugs in the interface. Added an ID field. Removed a couple unnecessary functions. Refactored the context check.

Be specific. Make it search friendly.

The worst message of all:

Fixed some bugs.

Hey wow, thanks for that.

Usually, this commit follows some previous larger commit that had a bunch of bugs. If so, say so! The lack of search-friendliness in the commit log can come back to haunt you. Be more descriptive. When you are skimming a commit log, or searching for a particular commit, these catch-all messages will be basically useless.

At the very least, name the important subsection/module of the code. “Fixed bad allocations in image processing routines” goes a long a way.

Communicate caveats

Bad:

Added super-widget 2.0

Better:

Very rough draft of super-widget 2.0. Compiles but is completely untested.

Tested + working super-widget 2.0... but the XYZ function() should be alot smarter to re-use previous computation

While your commit should never break anyone else’s build, it’s generally acceptable to commit things that aren’t “patched in” and aren’t necessarily fully working. Just state the important caveats in the commit. This is especially true if you have major line items to do. The third example above states missing functionality for others to see (and possibly implement), and it provides fantastic information when looking at the log for a particular file or module.

Blame someone else

If you looked in our code base, you might see commit messages like this, from me:

Fixed a huge memory leak in Mike’s code (which was cleverly hidden in the code I comitted yesterday... he is very crafty)

Place a bounty

Here’s another example of a commit message I’ve made:

The most bizarre commit ever. It obviously changes the code. It also obviously makes the code faster in my tests. The problem is that it makes absolutely no sense, whatsoever. \$5 to whomever convinces me that this works for some rational reason as opposed to my current

theory: a combination of black magic and alien technology.

This has a happy ending as someone was able to come up with a plausible explanation for what was going on. Like almost all speed improvements, it was due to cache. You can find the [nitty gritty details over here](#).

[trackback](#)

11 Responses to “Writing better commit messages”

1. [March 30th, 2009 at 12:26 pm](#)

 *Slappy* says:

Did you end up paying the \$5?

2. [March 30th, 2009 at 10:28 pm](#)

 *angelo* says:

In my commits, i always like to do stuff like


fix: database wasn't closing the connection right

add: implemented the call() method for database

...

I think if every developer committing changes followed a pattern like this, you'd be able to find logs easier.

3. [March 30th, 2009 at 10:37 pm](#)

 *ph0rque* says:

Is it bad that some of my commits are, “I don't remember what I did here...”

4. [March 30th, 2009 at 10:48 pm](#)


 *Josh Davis* says:

Heck yeah dude, good comment messages ROCK!

RT

<http://www.anonymity.us.tc>

5. [March 30th, 2009 at 11:05 pm](#)

 *bowl* says:

My favorite ones are the guys that constantly use “interim commit” as their sole message. Thanks dude!

6. [March 30th, 2009 at 11:27 pm](#)

 [Jake Boxer](#) says:

Back at the place I interned over the summer, one guy used the same message every time he committed: “Updated code”.

7. [March 31st, 2009 at 2:16 am](#)

 [Binny V A](#) says:

Great advice. I’ll try to follow them next time onwards.

@angelo That’s a good convention. Thanks for the tip.

8. [March 31st, 2009 at 3:58 pm](#)

 [Robert Cowham](#) says:

Some good thoughts.

Chris Seiwald (founder of <http://www.perforce.com> and an author of a chapter in “Beautiful Code”) gave a keynote in which he recommended:

- restrict all whitespace changes to a specific type and document as such
- restrict refactorings (that should not change functionality) to their own commit (and document)
- restrict functionality changes (bug fix or new feature) to their own changelists (and document)

i.e. don’t mix different types of change, and put a standard prefix for each type of change to identify it in your commit message.

Excellent idea (not always easy to restrict oneself to!).

Robert

9. [March 31st, 2009 at 6:26 pm](#)

 [Bill King](#) says:

As our repository is going public in a matter of days/weeks, may I add, don’t write something you wouldn’t want seen in public. We’ve chopped our commit history to a week ago, for privacy law reasons as much as anything


else, but still.... there were some pretty gruesome commit messages, and this is only in the 4.x stream.

10. [March 31st, 2009 at 9:23 pm](#)

 [Ryan Kohn](#) says:

The trick to a good commit message is thinking about how it will be used by someone else, or even yourself, in the future. Empathy goes a long way in building a good team.

11. [April 1st, 2009 at 2:54 pm](#)

 *louis* says:

@Bill King,

I have a few private repos that I just looked at the git/svn logs for just to see what I'd think about opening them up for the world to see. It wasn't so good, heh. I'll need to keep that one in mind.

Leave a Reply

Name

Mail (optional, never published)

Website (optional)

We need to make sure you are a human. Please solve the challenge below, and click the I'm a Human button to get a confirmation code. To make this process easier in the future, we recommend you enable Javascript.

cloths

Norton

Type the two words:



I'm a Human

Submit Comment

- [home](#)
- [about](#)
- [full archive](#)
- [feed](#)

Need a new job?

- © 2008 Louis Brandy. All Rights Reserved.
- Contact: **louis@lbrandy.com**
-