



# How to Become a Doc Committer

BY BENEDICT REUSCHLING



The FreeBSD documentation set is comprised of various handbooks for users and developers, FAQs, articles on specific topics, man pages in the src repository, and the main FreeBSD.org website. It is a great way for newcomers to make their first contribution to the FreeBSD Project and has been the gateway to many developers still active. While reading our documentation, people often find things that need correction: from a missing comma, to a typo in a word, to a whole paragraph that is outdated and needs rewriting. Those who do not shrug their shoulders and move on either report the bug on our Bugzilla [[https://bugs.freebsd.org/bugzilla/enter\\_bug.cgi?product=Documentation](https://bugs.freebsd.org/bugzilla/enter_bug.cgi?product=Documentation)] system or come up with a patch to fix the issue. While the latter is certainly not a requirement (especially for drive-by reports and people who do not yet know our doc toolchain), the former is still better than doing nothing about it.

It's also not just about finding bugs in existing material. If you have found an easier way to do something and can explain it, submit a bug report with your work. Other people may find that our handbook could do with more pictures to illustrate the concepts described there (after all, a picture is worth a thousand words) and may want to send us their sketches. Updated screenshots also fall into this category and help newcomers figure out that they are on the right track by comparing them with what's on their screen. Flyers and cheat sheets with common FreeBSD commands to hand out at conferences are another way to show your talent. Did I mention that we could use more EXAMPLE sections in various user man pages? Run link checkers on a regular basis on web pages we link to and report dead links (even better, find an alternative location). If you like wikis and make edits there, join the #freebsd-wiki group on the Freenode IRC. In other words, surprise us (in a good way) and show up with some examples to demonstrate that you are serious about your endeavor.

Whatever it is, a documentation committer will pick up your work (typically a patch or a textual description) and start working on it. If not, a gentle nudge may help, as developers are dealing with a lot of work. In the simplest case (comma or typo fix), they'll make changes to the sources directly and commit it, giving you as the original submitter the credit in the "Submitted by:" line of the commit message. Thank you for your contribution, you just made the FreeBSD documentation a little better!

In other instances (when it gets a bit more complicated or the patch is bigger), you can also create an account on our Phabricator instance [<https://reviews.freebsd.org/>] and submit the patch that way (adding the docs group or individuals as reviewers). This makes the patch review process easier to track. It allows feedback from various angles (writing style, formatting, etc.) and updates of an existing patch. Don't get frustrated if you only get a few lines of feedback that may sound

overly critical. In most cases, I've found that people who do reviews have only a few minutes to do these and thus keep their feedback short and simple. Don't be afraid to ask follow-up questions if you don't understand any of the jargon we use or don't know how to proceed. Don't take feedback as personal as it is not you we criticize—just the code. I typically try to mix some positive feedback into my comments. "Thank you for the patch," "Good catch," and "Glad you took the time to report this" can go a long way toward making a submitter feel good and willing to work the provided feedback into their code for another review round.

With time, patience, and some practice, you'll get your first contribution into the documentation tree. It feels good, right? You'll probably show this to your friends and colleagues, post it on social media, shout it from the highest mountain. I think people want these successes in their lives where they can point at something and proudly say "there, that's what I made." While some may say: "Huh, big deal," remember that it is based on your contribution that a committer took up this patch and made a change in the FreeBSD repository. You've become part of project history, a project whose roots go back over 30 years and counting. It's also a great way to work with people on something you care about. Many people like this so much that they continue to contribute other patches. And that's where the trouble begins...

Over time, the same person may submit a lot of good additions to the documentation tree. From a flood of small patches over a couple of weeks to groundbreaking, new work adding much more material to our documentation (and anything in between). People in the project start noticing. In addition, you take feedback to heart and work with developers in a constructive and collaborative way. Then one day it happens: the committer gets annoyed, because they are doing all this work for you that you could basically do yourself. That is what we say when we "punish" someone with a commit bit.

The doc committer will create a list of your past deeds (some may call it a special form of a "naughty list"). That is why we collect your track record in the "Submitted by:" field of the commit message. Phabricator can also generate a nice list of all your review interactions. This list is then sent to the Documentation Engineering Team (abbrv. doceng), which is (among other things) responsible for managing the documentation commit bits. A usual plea of "please let me mentor that person so that I can get my time back while they commit their own work" goes along with it. This proposal is then evaluated and voted on internally. Sometimes committers form a small tag team to better cover different time zones between mentor and mentee (faster feedback loops). The doceng team can pick additional mentors if they deem it necessary. If the vote comes out in favor of the mentee, they are given notice and granted access to Project resources necessary to work with the documentation. If the commit bit is not granted, we usually provide feedback on things that are missing (usually "beef up your track record") and encourage another proposal again after some time has passed.

The mentor will then teach the new mentee everything they need to know: from understanding our formatting rules as well as bumping the .Dd (document date) of a man page when there is a significant enough change being made, to their first nervous (often sweaty-handed) commit. During this time, the mentor will help the mentee when they make a mistake. The mentoring phase is meant to give the mentee enough confidence to not make these mistakes again and learn from the vast experience the mentor has from working with the Project. Social interactions from the mentee within the community are also a part of the process as well as welcoming feed-

*With time, patience, and some practice, you'll get your first contribution into the documentation tree.*

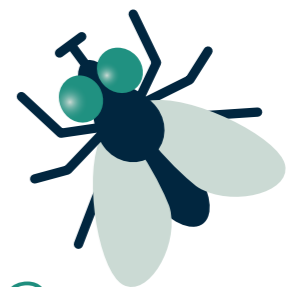
back from contributors (which completes the circle), working with other members of the community, and oftentimes committing their own submitted patches after mentor review.

Modern tooling like Phabricator makes this a much easier process than doing it over email, which works (that's how I was mentored way back when), but it takes longer. It's some intense work, but in my experience, also a lot of fun. I've seen mentors and mentees form a deep personal relationship from the process, which is a satisfying experience in itself. Each mentee is different, of course, but more often than not, I've seen a mentor and mentee meet at a conference for the first time and immediately talk like they've been old friends.

Some people in the project don't think it is worthwhile to mentor because it takes too much time away from their own FreeBSD work. That may be true, although I've found that the reduction in my work is paid back multiple times in added productivity for the Project as a whole. After all, two people are making changes and the passion and enthusiasm of a new mentee might motivate you to take on that long-dreaded documentation change together. Mentoring is not for everyone and so no one is forced to mentor. Luckily, there are enough people willing to do that and some are willing to co-mentor someone to share the load. In my experience, it's a tribe that mentors someone and not a single person. Feedback and encouragement are given not just by the mentor alone, which encourages the mentee to cast a wider net over time. I've seen many former mentees of mine move into other areas of the Project where I wouldn't go because my knowledge in that space is limited. But what a great benefit for the Project as a whole.

Be that as it may, the mentoring period may take a while, depending on the amount of work available, time constraints, and engagement of both mentee and mentor. After a while, mentors feel that they just approve changes without the need to do much correcting anymore and that the mentee has learned all there is to know. With the agreement of the mentee, they are released and are then an equal documentation committer like all the others. I personally encourage my former mentees in my passing message to continue to ask me questions if they are unsure. Also, I tell them the wise words of Yoda to "pass on what you have learned," meaning that they should take their own mentee when the time comes. I have often mentored other people with former mentees of mine, and have seen that they put the same kind of effort into the mentoring process.

Did I get you interested? Take a look at the FreeBSD Documentation Project Primer for New Contributors [[https://www.freebsd.org/doc/en\\_US.ISO8859-1/books/fdp-primer/](https://www.freebsd.org/doc/en_US.ISO8859-1/books/fdp-primer/)]. It shows you how to check out the documentation tree and how to make some changes to existing docs. The Quick Start section has these things covered. If you are interested in man-page changes, it has its own section. Other parts deal with formatting, formatting elements, and how to properly configure your editor to make your life a little easier. It's not too difficult to get started and who knows where this will lead to over time?



I tell them the wise words of Yoda to "pass on what you have learned."

---

**BENEDICT REUSCHLING** is a documentation committer in the FreeBSD project and member of the documentation engineering team. He serves on the board of directors of the FreeBSD Foundation as vice president. In the past, he served on the FreeBSD core team for two terms. He administers a big data cluster at the University of Applied Sciences, Darmstadt, Germany. He's also teaching a course "Unix for Developers" for undergraduates. Together with Allan Jude, he is host of the weekly [bsdnow.tv](https://bsdnow.tv) podcast.