

A simple bash script to automatically generate a CHANGELOG.md file with every git commit

How to use it

Copy the script in to a file `post-commit` and copy the file into the directory `.git/hooks/` of your git project and make it executable.

Customization

Generated file name

You can change the name of the generated CHANGELOG file inside the `post-commit` file by modifying the variable `OUTPUT_FILE` to whatever you want. Some projects also use `HISTORY.md`, `NEWS.md` or `RELEASE.md`, but I suggest you stick to the convention.

Output format

The default setting of the script generates markdown, which can be seen in the `CHANGELOG.md`.

If you want to change the output format, you should read the [Git documentation](#) for pretty-formats and change the `--format` argument in `post-commit` file according to your desired format.

You can test your output format by executing the following command inside your git repository folder:

```
git --no-pager log --no-merges --format="YOUR_FORMAT_GOES_HERE"
```

If you are happy with the result, just replace the command in the `post-commit` file with your command. Don't forget to pipe the output in you're `CHANGELOG` with `> $OUTPUT_FILE` at the end of the line.

How it works

The script uses [Git Hooks](#) to execute the `git log` command after every commit and write the output inside the `CHANGELOG.md` file.

Since we want to add the changes in our `CHANGELOG.md` as well, we have to use [Git's --amend feature](#) to change the previous commit and apply the changes from the script, too.

By changing the last commit, we are triggering the `post-commit` hook another time, since the

–amend commit is a commit as well. To prevent an infinite execution of this hook, we have to check, whether our generated CHANGELOG.md have have changed at all. If it is executed the second time, the file should not be different.

We can check this by using the [git status command with --porcelain](#) and parse the output for our \$OUTPUT_FILE. If the process returns a code higher than zero, we had a match and know that we have a changed CHANGELOG.md, which we have to append to our previous commit.

```
#!/bin/bash
set -eu

# destination of the final changelog file
OUTPUT_FILE=CHANGELOG.md

# generate the changelog
git --no-pager log --no-merges --format="### %s%d%n>%aD%n%n>Author: %aN (%aE)%n%n>Committer: %cN (%cE)%n%n%b%n%N%n" > $OUTPUT_FILE

# prevent recursion!
# since a 'commit --amend' will trigger the post-commit script again
# we have to check if the changelog file has changed or not
res=$(git status --porcelain | grep -c ".$OUTPUT_FILE$")
if [ "$res" -gt 0 ]; then
    git add $OUTPUT_FILE
    git commit --amend
    echo "Populated Changelog in $OUTPUT_FILE"
fi
```

Alle commits seit dem letzten Tag und ohne merges

```
git --no-pager log $(git describe --tags --abbrev=0)..HEAD --no-merges --
format="### %s%d%n>%aD%n%n>Author: %aN (%aE)%n%n>Committer: %cN (%cE)%n%n%b%n%N%n" > $OUTPUT_FILE
```

From:
<https://blog.cooltux.net/> - TuxNet DokuWiki

Permanent link:
https://blog.cooltux.net/doku.php?id=it-wiki:git:automatically_generate_changelog_md_file_with_git_commit

Last update: 2023/01/07 18:09

