

Git With Other Default Branch

Christian Külker

2023-05-04

Contents

1	Default Git Branch	1
2	Manually Changing Default Branch At Creation Time	1
3	Setting Globally a Different Branch Per Default	3
3.1	Example	4
4	Changing The Branch in Existing Repositories	4
5	History	5
6	Disclaimer of Warranty	5
7	Limitation of Liability	5

1 Default Git Branch

There is no way I know of to query the default branch name that git will use for a new repository. As of 2023-05-04 it is `master`.

2 Manually Changing Default Branch At Creation Time

Sometimes the default branch name is not what you want. However, this branch name is hard-coded into `git`. One way to get around using the hard-coded default branch is to create a virtual new branch **before** the first commit.

When a new `git` repository is created, the `HEAD` file is at that moment a pointer to the (virtual) default branch. You can easily find out what it points to with `cat`. The trick is

to change the pointer for HEAD **before** the first commit, and therefore before the default branch exists, and make the new virtual first branch the default branch.

For all steps, we will assume that a new repository has been created, similar to this approach.

```
cd /tmp
mkdir test
cd test
git init
```

(VS;PR) Now the **very short** process of creating a git repository with a different default branch, without explanations, so **please read** it.

```
cd /tmp
mkdir test
cd test
git init
git checkout --orphan main
```

(ATL;DR) The next example is basically (almost) the same, but includes some commands to prove it and explanations of what to expect.

```
cd /tmp
mkdir test
cd test
git init # HEAD points to non existing ref:
↳ refs/heads/master
cat .git/HEAD # Should give ref: refs/heads/master
ls .git/refs/heads # Should give nothing
git checkout --orphan main # HEAD points to non existing ref:
↳ refs/heads/main
cat .git/HEAD # Should give ref: refs/heads/main
ls .git/refs/heads # Should give nothing (will change after commit)
```

(TL;DR) Finally, the following session shows what happens when there is a first commit to the repository, including the output of all commands.

```
cd /tmp
mkdir test
cd test

git init # HEAD points to non existing ref:
↳ refs/heads/master
```

```
Initialized empty Git repository in /tmp/test/.git/

cat .git/HEAD          # Should give ref: refs/heads/master
ref: refs/heads/master

ls .git/refs/heads     # Should give nothing

git checkout --orphan main # HEAD points to non existing ref:
↪ refs/heads/main
Switched to a new branch 'main'

cat .git/HEAD          # Should give ref: refs/heads/main
ref: refs/heads/main

ls .git/refs/heads     # Should give nothing (will change after commit)

touch README.md        # Will create the file README.md

git add README.md      # Make the file known to git

                        # Create a commit and in unison the new branch
                        # under a non default name
git commit -m init README.md
[main (root-commit) 9fdf194] init
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md

cat .git/HEAD          # Should give ref: refs/heads/main
ref: refs/heads/main

ls .git/refs/heads     # Should give main
main
```

As you can see, `ls .git/refs/heads` now shows that the new branch has been created.

3 Setting Globally a Different Branch Per Default

The previous section described the situation where you are not happy with the default branch name. As of Git 2.28.0, release July 2020, it is possible to use the `init.defaultBranch` configuration option, which allows Git users to define and

configure a default branch name other than `master`.

For example, Gitlab changed the default branch from `master` to `main` in 2021.

3.1 Example

To change the default branch name in Git from `master` to `bonanza` (for example) for all new repositories, you can use the `init.defaultBranch` configuration setting. This setting was introduced in Git 2.28.0, so make sure you have a compatible version installed.

You can check your Git version using the following command:

```
git --version
```

If you have a compatible version, you can use this command to set the default branch name:

```
git config --global init.defaultBranch bonanza
```

This will set the default branch name to `bonanza` for all new repositories created by the current user. If you want to set it for a specific repository only, you should navigate to that repository and run this command without the `--global` flag:

```
git config init.defaultBranch bonanza
```

4 Changing The Branch in Existing Repositories

Note that the above receipt will not change the default branch name in existing repositories. For existing repositories you will need to manually rename the branch. Here is how to do it:

1. Switch to the branch you want to rename:

```
git checkout master
```

2. Rename the branch:

```
git branch -m bonanza
```

3. Push the renamed branch and reset the upstream branch:

```
git push origin -u bonanza
```

4. Finally, you may want to delete the old branch in the remote repository:

```
git push origin --delete master
```

Remember to inform your staff/community/co-workers about this change, as it will affect their clones.

5 History

Version	Date	Notes
0.1.2	2023-05-04	Improve writing, Git 2.80.0, init.defaultBranch
0.1.1	2023-05-03	Improve writing
0.1.0	2022-07-15	Initial release

6 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOCUMENT OR THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

7 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.