
git-new-branch(1) Manual Page

NAME

`git-new-branch` - Create a new branch with correct tracking information.

SYNOPSIS

```
git new-branch <branch_name>
git new-branch --upstream_current <branch_name>
git new-branch --upstream <REF> <branch_name>
git new-branch --lkgr <branch_name>
```

DESCRIPTION

Creates a new branch. By default the new branch will track the configured upstream for the repo (defaults to *origin/master*). If one of the other options is specified, it will track that other ref instead.

Conceptually, each branch in your repo represents one *Change List (CL)*. If you have many independent CLs (i.e. the changes in one do not interact with/depend on the changes in another), then you should create them as new branches tracking the default upstream (i.e. `git new-branch <branch_name>`). If you have features which depend on each other, you should create stacked branches using `git new-branch --upstream_current <branch_name>`.

OPTIONS

`--upstream_current`

Set the tracking (upstream) branch to the currently-checked-out branch.

`--upstream <REF>`

Set the tracking (upstream) branch to <REF>. <REF> may be a local branch, remote branch, or a tag.

`--lkgr`

Alias for `--upstream lkgr`.

`<branch_name>`

The name for the new branch.

CONFIGURATION VARIABLES

depot-tools.upstream

This configures the default *upstream* for all new branches. If it is unset, it defaults to *origin/master*. This is considered to be the *root* branch.

EXAMPLE

```
$ git map-branches -v
origin/master
  cool_feature          [ ahead 4 ]
  subfeature            [ ahead 2 | behind 1 ]
  fixit                 [ ahead 2 ]
  frozen_branch *       [ ahead 3 ]
$ git new-branch independent_cl
$ git map-branches -v
origin/master
  cool_feature          [ ahead 4 ]
  subfeature            [ ahead 2 | behind 1 ]
  fixit                 [ ahead 2 ]
  frozen_branch         [ ahead 3 ]
  independent_cl *      [ ahead 3 ]
$ vi foo && git add -A && git commit -m foo
$ git map-branches -v
origin/master
  cool_feature          [ ahead 4 ]
  subfeature            [ ahead 2 | behind 1 ]
  fixit                 [ ahead 2 ]
  frozen_branch         [ ahead 3 ]
  independent_cl *      [ ahead 1 ]
$ git new-branch --upstream subfeature nested_cl
$ git map-branches -v
origin/master
```

```

cool_feature          [ ahead 4 ]
  subfeature          [ ahead 2 | behind 1 ] <1>
    nested_cl *
  fixit               [ ahead 2 ]
    frozen_branch     [ ahead 3 ]
  independent_cl      [ ahead 1 ]
$ git checkout cool_feature
Switched to branch 'cool_feature'
Your branch is ahead of 'origin/master' by 4
commits.
(use "git push" to publish your local commits)
$ git new-branch --upstream_current
cl_depends_on_cool_feature
$ git map-branches -v
origin/master
  cool_feature          [ ahead 4 ]
]
  cl_depends_on_cool_feature *
  subfeature            [ ahead 2 |
behind 1 ]
    nested_cl
  fixit                 [ ahead 2 ]
]
  frozen_branch         [ ahead 3 ]
]
  independent_cl        [ ahead 1 ]
]

```

1. Note that both branches are cyan because they are currently the same *commit* object. See [git-map-branches\(1\)](#) for more detail.

SUGGESTED ALIASES

Some common short-hand aliases. Feel free to add these to your `~/.gitconfig` file.

```

[alias]
git nb = new-branch
git tb = new-branch --upstream_current <1>

```

1. mnemonic: tb → "track branch"

SEE ALSO

[git-rebase-update\(1\)](#), [git-reparent-branch\(1\)](#), [git-rename-branch\(1\)](#), [git-upstream-diff\(1\)](#).