branch

`branch.autoSetupMerge`
Tells `git branch`, `git switch` and `git checkout` to set up new branches so that linkgit:git-pull[1] will appropriately merge from the starting point branch. Note that even if this option is not set, this behavior can be chosen per-branch using the `--track` and `--no-track` options. This option defaults to `true`. The valid settings are: `false`;; no automatic setup is done `true`;; automatic setup is done when the starting point is a remote-tracking branch `always`;; automatic setup is done when the starting point is either a local branch or remote-tracking branch `inherit`;; if the starting point has a tracking configuration, it is copied to the new branch `simple`;; automatic setup is done only when the starting point is a remote-tracking branch and the new branch has the same name as the remote branch.

`branch.autoSetupRebase`
When a new branch is created with `git branch`, `git switch` or `git checkout` that tracks another branch, this variable tells Git to set up pull to rebase instead of merge (see `branch.<name>.rebase`). The valid settings are: `never`;; rebase is never automatically set to true. `local`;; rebase is set to true for tracked branches of other local branches. `remote`;; rebase is set to true for tracked branches of remote-tracking branches. `always`;; rebase will be set to true for all tracking branches.

See `branch.autoSetupMerge` for details on how to set up a branch to track another branch. This option defaults to `never`.

`branch.sort`
This variable controls the sort ordering of branches when displayed by linkgit:git-branch[1]. Without the `--sort=<value>` option provided, the value of this variable will be used as the default. See linkgit:git-for-each-ref[1] field names for valid values.

`branch.<name>.remote`
When on branch <*name*>, it tells `git fetch` and `git push` which remote to fetch from or push to. The remote to push to may be overridden with `remote.pushDefault` (for all branches). The remote to push to, for the current branch, may be further overridden by `branch.<name>.pushRemote`. If no remote is configured, or if you are not on any branch and there is more than one remote defined in the repository, it defaults to `origin` for fetching and `remote.pushDefault` for pushing. Additionally, `.` (a period) is the current local repository (a dot-repository), see `branch.<name>.merge`'s final note below.

`branch.<name>.pushRemote`
When on branch <*name*>, it overrides `branch.<name>.remote` for pushing. It also overrides `remote.pushDefault` for pushing from branch <*name*>. When you pull from one place (e.g. your upstream) and push to another place (e.g. your own publishing repository), you would want to set `remote.pushDefault`

to specify the remote to push to for all branches, and use this option to override it for a specific branch. The value may be the name of a configured remote or a repository URL.

`branch.<name>.merge`
Defines, together with `branch.<name>.remote`, the upstream branch for the given branch. It tells `git` `fetch`/`git` `pull`/`git` `rebase` which branch to merge and can also affect `git` `push` (see `push.default`). When in branch <*name*>, it tells `git` `fetch` the default refspec to be marked for merging in `FETCH_HEAD`. The value is handled like the remote part of a refspec, and must match a ref which is fetched from the remote given by `branch.<name>.remote`. The merge information is used by `git` `pull` (which first calls `git` `fetch`) to lookup the default branch for merging. Without this option, `git` `pull` defaults to merge the first refspec fetched. Specify multiple values to get an octopus merge. If you wish to setup `git` `pull` so that it merges into <*name*> from another branch in the local repository, you can point `branch.<name>.merge` to the desired branch, and use the relative path setting `.` (a period) for `branch.<name>.remote`.

`branch.<name>.mergeOptions`
Sets default options for merging into branch <*name*>. The syntax and supported options are the same as those of linkgit:git-merge[1], but option values containing whitespace characters are currently not supported.

`branch.<name>.rebase`
When true, rebase the branch <*name*> on top of the fetched branch, instead of merging the default branch from the default remote when `git` `pull` is run. See `pull.rebase` for doing this in a non branch-specific manner.

When `merges` (or just `m`), pass the `--rebase-merges` option to `git` `rebase` so that the local merge commits are included in the rebase (see linkgit:git-rebase[1] for details).

When the value is `interactive` (or just `i`), the rebase is run in interactive mode.

NOTE: this is a possibly dangerous operation; do **not** use it unless you understand the implications (see linkgit:git-rebase[1] for details).

`branch.<name>.description`
Branch description, can be edited with `git` `branch` `--edit-description`. Branch description is automatically added to the `format-patch` cover letter or `request-pull` summary.

`branch.<name>.deleteMerged`
If set to `false`, branch <*name*> is exempt from `git` `branch` `--delete-merged`. Useful for a topic branch you intend to develop further after an initial round has been merged upstream. Defaults to true. Explicit deletion via `git` `branch` `-d` is unaffected.

□□□ #2
□ □□□ □□□ 2026-10-31 20:06:04 UTC
□ □□□ □□□ 2026-10-31 20:06:06 UTC