advice

advice.*
These variables control various optional help messages designed to aid new users. When left unconfigured, Git will give the message alongside instructions on how to squelch it. You can tell Git that you have understood the issue and no longer need a specific help message by setting the corresponding variable to `false`.

As they are intended to help human users, these messages are output to the standard error. When tools that run Git as a subprocess find them disruptive, they can set `GIT_ADVICE=0` in the environment to squelch all advice messages.

addEmbeddedRepo
Shown when the user accidentally adds one git repo inside of another.

addEmptyPathspec
Shown when the user runs `git` `add` without providing the pathspec parameter.

addIgnoredFile
Shown when the user attempts to add an ignored file to the index.

amWorkDir
Shown when linkgit:git-am[1] fails to apply a patch file, to tell the user the location of the file.

ambiguousFetchRefspec
Shown when a fetch refspec for multiple remotes maps to the same remote-tracking branch namespace and causes branch tracking set-up to fail.

checkoutAmbiguousRemoteBranchName
Shown when the argument to linkgit:git-checkout[1] and linkgit:git-switch[1] ambiguously resolves to a remote tracking branch on more than one remote in situations where an unambiguous argument would have otherwise caused a remote-tracking branch to be checked out. See the `checkout.defaultRemote` configuration variable for how to set a given remote to be used by default in some situations where this advice would be printed.

commitBeforeMerge
Shown when linkgit:git-merge[1] refuses to merge to avoid overwriting local changes.

detachedHead
Shown when the user uses linkgit:git-switch[1] or linkgit:git-checkout[1] to move to the detached HEAD state, to tell the user how to create a local branch after the fact.

diverging
Shown when a fast-forward is not possible.

fetchRemoteHEADWarn
Shown when linkgit:git-fetch[1] reveals that a remote `HEAD` differs from what is set locally and the user has opted into receiving a warning in this situation.

fetchShowForcedUpdates
Shown when linkgit:git-fetch[1] takes a long time to calculate forced updates after ref updates, or to warn that the check is disabled.

forceDeleteBranch
Shown when the user tries to delete a not fully merged branch without the force option set.

historyUpdateRefs
Shown when `git` `history` `squash` refuses because a local branch cannot be rewritten as a descendant of the squashed commit, to tell the user about `--update-refs=head`.

ignoredHook
Shown when a hook is ignored because the hook is not set as executable.

implicitIdentity
Shown when the user’s information is guessed from the system username and domain name, to tell the user how to set their identity configuration.

mergeConflict
Shown when various commands stop because of conflicts.

nestedTag
Shown when a user attempts to recursively tag a tag object.

pushAlreadyExists
Shown when linkgit:git-push[1] rejects an update that does not qualify for fast-forwarding (e.g., a tag.)

pushFetchFirst
Shown when linkgit:git-push[1] rejects an update that tries to overwrite a remote ref that points at an object we do not have.

pushNeedsForce
Shown when linkgit:git-push[1] rejects an update that tries to overwrite a remote ref that points at an object that is not a commit-ish, or make the remote ref point at an object that is not a commit-ish.

pushNonFFCurrent
Shown when linkgit:git-push[1] fails due to a non-fast-forward update to the current branch.

pushNonFFMatching
Shown when the user ran linkgit:git-push[1] and pushed "matching refs" explicitly (i.e. used `:`, or specified a refspec that isn’t the current branch) and it resulted in a non-fast-forward error.

pushRefNeedsUpdate
Shown when linkgit:git-push[1] rejects a forced update of a branch when its remote-tracking ref has updates that we do not have locally.

pushRepoLooksLikeRef
Shown when the repository given to linkgit:git-push[1] is not a configured remote but looks like a `<remote>/<branch>` ref, suggesting that the remote and branch be given as separate arguments.

pushUnqualifiedRefname
Shown when linkgit:git-push[1] gives up trying to guess based on the source and destination refs what remote ref namespace the source belongs in, but where we can still suggest that the user push to either `refs/heads/*` or `refs/tags/*` based on the type of the source object.

pushUpdateRejected
Set this variable to `false` if you want to disable `pushNonFFCurrent`, `pushNonFFMatching`, `pushAlreadyExists`, `pushFetchFirst`, `pushNeedsForce`, and `pushRefNeedsUpdate` simultaneously.

rebaseTodoError
Shown when there is an error after editing the rebase todo list.

refSyntax
Shown when the user provides an illegal ref name, to tell the user about the ref syntax documentation.

resetNoRefresh
Shown when linkgit:git-reset[1] takes more than 2 seconds to refresh the index after reset, to tell the user that they can use the `--no-refresh` option.

resolveConflict
Shown by various commands when conflicts prevent the operation from being performed.

rmHints
Shown on failure in the output of linkgit:git-rm[1], to give directions on how to proceed from the current state.

sequencerInUse
Shown when a sequencer command is already in progress.

skippedCherryPicks
Shown when linkgit:git-rebase[1] skips a commit that has already been cherry-picked onto the upstream branch.

sparseIndexExpanded
Shown when a sparse index is expanded to a full index, which is likely due to an unexpected set of files existing outside of the sparse-checkout.

statusAheadBehind
Shown when linkgit:git-status[1] computes the ahead/behind counts for a local ref compared to its remote tracking ref, and that calculation takes longer than expected. Will not appear if `status.aheadBehind` is false or the option `--no-ahead-behind` is given.

statusHints
Show directions on how to proceed from the current state in the output of linkgit:git-status[1], in the template shown when writing commit messages in linkgit:git-commit[1], and in the help message shown by linkgit:git-switch[1] or linkgit:git-checkout[1] when switching branches.

statusUoption
Shown when linkgit:git-status[1] takes more than 2 seconds to enumerate untracked files, to tell the user that they can use the `-u` option.

submoduleAlternateErrorStrategyDie
Shown when a submodule.alternateErrorStrategy option configured to "die" causes a fatal error.

submoduleMergeConflict
Advice shown when a non-trivial submodule merge conflict is encountered.

submodulesNotUpdated
Shown when a user runs a submodule command that fails because `git` `submodule` `update` `--init` was not run.

suggestDetachingHead
Shown when linkgit:git-switch[1] refuses to detach HEAD without the explicit `--detach` option.

updateSparsePath
Shown when either linkgit:git-add[1] or linkgit:git-rm[1] is asked to update index entries outside the current sparse checkout.

waitingForEditor
Shown when Git is waiting for editor input. Relevant when e.g. the editor is not launched inside the terminal.

worktreeAddOrphan
Shown when the user tries to create a worktree from an invalid reference, to tell the user how to create a new unborn branch instead.

□□□ #2
□ □□□ □□□ 2026-10-31 20:05:50 UTC
□ □□□ □□□ 2026-10-31 20:05:52 UTC