remote

remote.pushDefault
The remote to push to by default. Overrides `branch.<name>.remote` for all branches, and is overridden by `branch.<name>.pushRemote` for specific branches.

remote.<name>.url
The URL of a remote repository. See linkgit:git-fetch[1] or linkgit:git-push[1]. A configured remote can have multiple URLs; in this case the first is used for fetching, and all are used for pushing (assuming no `remote.<name>.pushurl` is defined). Setting this key to the empty string clears the list of urls, allowing you to override earlier config.

remote.<name>.pushurl
The push URL of a remote repository. See linkgit:git-push[1]. If a `pushurl` option is present in a configured remote, it is used for pushing instead of `remote.<name>.url`. A configured remote can have multiple push URLs; in this case a push goes to all of them. Setting this key to the empty string clears the list of urls, allowing you to override earlier config.

remote.<name>.proxy
For remotes that require curl (http, https and ftp), the URL to the proxy to use for that remote. Set to the empty string to disable proxying for that remote.

remote.<name>.proxyAuthMethod
For remotes that require curl (http, https and ftp), the method to use for authenticating against the proxy in use (probably set in `remote.<name>.proxy`). See `http.proxyAuthMethod`.

remote.<name>.fetch
The default set of "refspec" for linkgit:git-fetch[1]. See linkgit:git-fetch[1].

remote.<name>.push
The default set of "refspec" for linkgit:git-push[1]. See linkgit:git-push[1].

remote.<name>.mirror
If true, pushing to this remote will automatically behave as if the `--mirror` option was given on the command line.

remote.<name>.skipDefaultUpdate
A deprecated synonym to `remote.<name>.skipFetchAll` (if both are set in the configuration files with different values, the value of the last occurrence will be used).

remote.<name>.skipFetchAll
If true, this remote will be skipped when updating using linkgit:git-fetch[1], the `update` subcommand of linkgit:git-remote[1], and ignored by the prefetch task of `git` `maintenance`.

remote.<name>.receivepack
The default program to execute on the remote side when pushing. See option --receive-pack of linkgit:git-push[1].

remote.<name>.uploadpack
The default program to execute on the remote side when fetching. See option --upload-pack of linkgit:git-fetch-pack[1].

remote.<name>.tagOpt
Setting this value to --no-tags disables automatic tag following when fetching from remote <name>. Setting it to --tags will fetch every tag from remote <name>, even if they are not reachable from remote branch heads. Passing these flags directly to linkgit:git-fetch[1] can override this setting. See options --tags and --no-tags of linkgit:git-fetch[1].

remote.<name>.vcs
Setting this to a value <vcs> will cause Git to interact with the remote with the git-remote-<vcs> helper.

remote.<name>.prune
When set to true, fetching from this remote by default will also remove any remote-tracking references that no longer exist on the remote (as if the `--prune` option was given on the command line). Overrides `fetch.prune` settings, if any.

remote.<name>.pruneTags
When set to true, fetching from this remote by default will also remove any local tags that no longer exist on the remote if pruning is activated in general via `remote.<name>.prune`, `fetch.prune` or `--prune`. Overrides `fetch.pruneTags` settings, if any.

See also `remote.<name>.prune` and the PRUNING section of linkgit:git-fetch[1].

remote.<name>.promisor
When set to true, this remote will be used to fetch promisor objects.

remote.<name>.advertisedAs
When a promisor remote is automatically configured using information advertised by a server through the `promisor-remote` protocol capability (see `promisor.acceptFromServerUrl`), the server's originally advertised name is saved in this variable. This is for information, tracing and debugging purposes. Users should not typically modify or create such configuration entries.

remote.<name>.partialclonefilter
The filter that will be applied when fetching from this promisor remote. Changing or clearing this value will only affect fetches for new commits. To fetch associated objects for commits already present in the local object database, use the `--refetch` option of linkgit:git-fetch[1].

remote.<name>.serverOption
The default set of server options used when fetching from this remote. These server options can be overridden by the `--server-option=` command line arguments.

This is a multi-valued variable, and an empty value can be used in a higher priority configuration file (e.g. `.git/config` in a repository) to clear the values inherited from a lower priority configuration files (e.g. `$HOME/.gitconfig`).

remote.<name>.negotiationRestrict
When negotiating with this remote during `git` `fetch`, restrict the commits advertised as "have" lines to only those reachable from refs matching the given patterns. This multi-valued config option behaves like `--negotiation-restrict` on the command line.

Each value is either an exact ref name (e.g. `refs/heads/release`) or a glob pattern (e.g. `refs/heads/release/*`). The pattern syntax is the same as for `--negotiation-restrict`.

These config values are used as defaults for the `--negotiation-restrict` command-line option. If `--negotiation-restrict` (or its synonym `--negotiation-tip`) is specified on the command line, then the config values are not used.

These values also influence negotiation during `git` `push` if `push.negotiate` is enabled.

Blank values signal to ignore all previous values, allowing a reset of the list from broader config scenarios.

remote.<name>.negotiationInclude
When negotiating with this remote during `git` `fetch`, the client advertises a list of commits that exist locally. In repos with many references, this list of "haves" can be truncated. Depending on data shape, dropping certain references may be expensive. This multi-valued config option specifies references, commit hashes, or ref pattern globs whose tips should always be sent as "have" commits during fetch negotiation with this remote.

Each value is either an exact ref name (e.g. `refs/heads/release`), a commit hash, or a glob pattern (e.g. `refs/heads/release/*`). The pattern syntax is the same as for `--negotiation-include`.

These config values are used as defaults for the `--negotiation-include` command-line option. If `--negotiation-include` is specified on the command line, then the config values are not used.

This option is additive with the normal negotiation process: the negotiation algorithm still runs and advertises its own selected commits, but the refs matching `remote.<name>.negotiationInclude` are sent unconditionally on top of those heuristically selected commits.

These values also influence negotiation during `git` `push` if `push.negotiate` is enabled.

Blank values signal to ignore all previous values, allowing a reset of the list from broader config scenarios.

remote.<name>.followRemoteHEAD
When fetching this remote using its default refspec, this setting determines how to handle differences between the remote's `HEAD` and the local `remotes/<name>/HEAD` symbolic-ref. Overrides the value of `fetch.followRemoteHEAD`. See `fetch.followRemoteHEAD` for a description of accepted

values.

In addition to the values supported by `fetch.followRemoteHEAD`, this setting may also take on the value "warn-if-not-`$branch`", which behaves like "warn", but ignores the warning if the remote's `HEAD` is `remotes/<name>/$branch`.

□□□ #2
□ □□□ □□□ 2026-10-31 20:08:17 UTC
□ □□□ □□□ 2026-10-31 20:08:18 UTC