1.7.2

Git v1.7.2 Release Notes ========================

Updates since v1.7.1 --------------------

- core.eol configuration and text/eol attributes are the new way to control the end of line conventions for files in the working tree.

- core.autocrlf has been made safer - it will now only handle line endings for new files and files that are LF-only in the repository. To normalize content that has been checked in with CRLF, use the new eol/text attributes.

- The whitespace rules used in "git apply --whitespace" and "git diff" gained a new member in the family (tab-in-indent) to help projects with policy to indent only with spaces.

- When working from a subdirectory, by default, git does not look for its metadirectory ".git" across filesystems, primarily to help people who have invocations of git in their custom PS1 prompts, as being outside of a git repository would look for ".git" all the way up to the root directory, and NFS mounts are often slow. DISCOVERY_ACROSS_FILESYSTEM environment variable can be used to tell git not to stop at a filesystem boundary.

- Usage help messages generated by parse-options library (i.e. most of the Porcelain commands) are sent to the standard output now.

- ':/<string>' notation to look for a commit now takes regular expression and it is not anchored at the beginning of the commit log message anymore (this is a backward incompatible change).

- "git" wrapper learned "-c name=value" option to override configuration variable from the command line.

- Improved portability for various platforms including older SunOS, HP-UX 10/11, AIX, Tru64, etc. and platforms with Python 2.4.

- The message from "git am -3" has been improved when conflict resolution ended up making the patch a no-op.

- "git blame" applies the textconv filter to the contents it works on, when available.

- "git checkout --orphan newbranch" is similar to "-b newbranch" but prepares to create a root commit that is not connected to any existing commit.

- "git cherry-pick" learned to pick a range of commits (e.g. "cherry-pick A..B" and "cherry-pick --stdin"), so did "git revert"; these do not support the nicer sequencing control "rebase [-i]" has, though.

- "git cherry-pick" and "git revert" learned --strategy option to specify the merge strategy to be used when performing three-way merges.

- "git cvsserver" can be told to use pserver; its password file can be stored outside the repository.

- The output from the textconv filter used by "git diff" can be cached to speed up their reuse.

- "git diff --word-diff=<mode>" extends the existing "--color-words" option, making it more useful in color-challenged environments.

- The regexp to detect function headers used by "git diff" for PHP has been enhanced for visibility modifiers (public, protected, etc.) to better support PHP5.

- "diff.noprefix" configuration variable can be used to implicitly ask for "diff --no-prefix" behaviour.

- "git for-each-ref" learned "%(objectname:short)" that gives the object name abbreviated.

- "git format-patch" learned --signature option and format.signature configuration variable to customize the e-mail signature used in the output.

- Various options to "git grep" (e.g. --count, --name-only) work better with binary files.

- "git grep" learned "-Ovi" to open the files with hits in your editor.

- "git help -w" learned "chrome" and "chromium" browsers.

- "git log --decorate" shows commit decorations in various colours.

- "git log --follow <path>" follows across copies (it used to only follow renames). This may make the processing more expensive.

- "git log --pretty=format:<template>" specifier learned "% <something>" magic that inserts a space only when %<something> expands to a non-empty string; this is similar to "%+<something>" magic, but is useful in a context to generate a single line output.

- "git notes prune" learned "-n" (dry-run) and "-v" options, similar to what "git prune" has.

- "git patch-id" can be fed a mbox without getting confused by the signature line in the format-patch output.

- "git remote" learned "set-branches" subcommand.
- "git rev-list A..B" learned --ancestry-path option to further limit the result to the commits that are on the ancestry chain between A and B (i.e. commits that are not descendants of A are excluded).
- "git show -5" is equivalent to "git show --do-walk 5"; this is similar to the update to make "git show master..next" walk the history, introduced in 1.6.4.
- "git status [-s] --ignored" can be used to list ignored paths.
- "git status -s -b" shows the current branch in the output.
- "git status" learned "--ignore-submodules" option.
- Various "gitweb" enhancements and clean-ups, including syntax highlighting, "plackup" support for instaweb, .fcgi suffix to run it as FastCGI script, etc.
- The test harness has been updated to produce TAP-friendly output.
- Many documentation improvement patches are also included.

Fixes since v1.7.1 ------------------

All of the fixes in v1.7.1.X maintenance series are included in this release, unless otherwise noted.

- We didn’t URL decode "file:///path/to/repo" correctly when path/to/repo had percent-encoded characters (638794c, 9d2e942, ce83eda, 3c73a1d).
- "git clone" did not configure remote.origin.url correctly for bare clones (df61c889).
- "git diff --graph" works better with "--color-words" and other options (81fa024..4297c0a).
- "git diff" could show ambiguous abbreviation of blob object names on its "index" line (3e5a188).
- "git reset --hard" started from a wrong directory and a working tree in a nonstandard location is in use got confused (560fb6a1).
- "git read-tree -m A B" used to switch to branch B while retaining local changes added an incorrect cache-tree information (b1f47514).

□□□ #2
□ □□□ □□□ 2026-10-31 19:49:55 UTC
□ □□□ □□□ 2026-10-31 19:49:56 UTC