1.6.1

GIT v1.6.1 Release Notes ==========

Updates since v1.6.0 ----------

When some commands (e.g. "git log", "git diff") spawn pager internally, we used to make the pager the parent process of the git command that produces output. This meant that the exit status of the whole thing comes from the pager, not the underlying git command. We swapped the order of the processes around and you will see the exit code from the command from now on.

(subsystems)

- gitk can call out to git-gui to view "git blame" output; git-gui in turn can run gitk from its blame view.
- Various git-gui updates including updated translations.
- Various gitweb updates from repo.or.cz installation.
- Updates to emacs bindings.

(portability)

- A few test scripts used nonportable "grep" that did not work well on some platforms, e.g. Solaris.
- Sample pre-auto-gc script has OS X support.
- Makefile has support for (ancient) FreeBSD 4.9.

(performance)

- Many operations that are lstat(3) heavy can be told to pre-execute necessary lstat(3) in parallel before their main operations, which potentially gives much improved performance for cold-cache cases or in environments with weak metadata caching (e.g. NFS).
- The underlying diff machinery to produce textual output has been optimized, which would result in faster "git blame" processing.
- Most of the test scripts (but not the ones that try to run servers) can be run in parallel.
- Bash completion of refnames in a repository with massive number of refs has been optimized.

- Cygwin port uses native stat/lstat implementations when applicable, which leads to improved performance.
- "git push" pays attention to alternate repositories to avoid sending unnecessary objects.
- "git svn" can rebuild an out-of-date rev_map file.

(usability, bells and whistles)

- When you mistype a command name, git helpfully suggests what it guesses you might have meant to say. help.autocorrect configuration can be set to a non-zero value to accept the suggestion when git can uniquely guess.
- The packfile machinery hopefully is more robust when dealing with corrupt packs if redundant objects involved in the corruption are available elsewhere.
- "git add -N path..." adds the named paths as an empty blob, so that subsequent "git diff" will show a diff as if they are creation events.
- "git add" gained a built-in synonym for people who want to say "stage changes" instead of "add contents to the staging area" which amounts to the same thing.
- "git apply" learned --include=paths option, similar to the existing --exclude=paths option.
- "git bisect" is careful about a user mistake and suggests testing of merge base first when good is not a strict ancestor of bad.
- "git bisect skip" can take a range of commits.
- "git blame" re-encodes the commit metainfo to UTF-8 from i18n.commitEncoding by default.
- "git check-attr --stdin" can check attributes for multiple paths.
- "git checkout --track origin/hack" used to be a syntax error. It now DWIMs to create a corresponding local branch "hack", i.e. acts as if you said "git checkout --track -b hack origin/hack".
- "git checkout --ours/--theirs" can be used to check out one side of a conflicting merge during conflict resolution.
- "git checkout -m" can be used to recreate the initial conflicted state during conflict resolution.
- "git cherry-pick" can also utilize rerere for conflict resolution.
- "git clone" learned to be verbose with -v

- "git commit --author=$name" can look up author name from existing commits.
- output from "git commit" has been reworded in a more concise and yet more informative way.
- "git count-objects" reports the on-disk footprint for packfiles and their corresponding idx files.
- "git daemon" learned --max-connections=<count> option.
- "git daemon" exports REMOTE_ADDR to record client address, so that spawned programs can act differently on it.
- "git describe --tags" favours closer lightweight tags than farther annotated tags now.
- "git diff" learned to mimic --suppress-blank-empty from GNU diff via a configuration option.
- "git diff" learned to put more sensible hunk headers for Python, HTML and ObjC contents.
- "git diff" learned to vary the a/ vs b/ prefix depending on what are being compared, controlled by diff.mnemonicprefix configuration.
- "git diff" learned --dirstat-by-file to count changed files, not number of lines, when summarizing the global picture.
- "git diff" learned "textconv" filters --- a binary or hard-to-read contents can be munged into human readable form and the difference between the results of the conversion can be viewed (obviously this cannot produce a patch that can be applied, so this is disabled in format-patch among other things).
- "--cached" option to "git diff has an easier to remember synonym "--staged", to ask "what is the difference between the given commit and the contents staged in the index?"
- "git for-each-ref" learned "refname:short" token that gives an unambiguously abbreviated refname.
- Auto-numbering of the subject lines is the default for "git format-patch" now.
- "git grep" learned to accept -z similar to GNU grep.
- "git help" learned to use GIT_MAN_VIEWER environment variable before using "man" program.
- "git imap-send" can optionally talk SSL.
- "git index-pack" is more careful against disk corruption while completing a thin pack.

- "git log --check" and "git log --exit-code" passes their underlying diff status with their exit status code.
- "git log" learned --simplify-merges, a milder variant of --full-history; "gitk --simplify-merges" is easier to view than with --full-history.
- "git log" learned "--source" to show what ref each commit was reached from.
- "git log" also learned "--simplify-by-decoration" to show the birds-eye-view of the topology of the history.
- "git log --pretty=format:" learned "%d" format element that inserts names of tags that point at the commit.
- "git merge --squash" and "git merge --no-ff" into an unborn branch are noticed as user errors.
- "git merge -s $strategy" can use a custom built strategy if you have a command "git-merge-$strategy" on your $PATH.
- "git pull" (and "git fetch") can be told to operate "-v"erbosely or "-q"uietly.
- "git push" can be told to reject deletion of refs with receive.denyDeletes configuration.
- "git rebase" honours pre-rebase hook; use --no-verify to bypass it.
- "git rebase -p" uses interactive rebase machinery now to preserve the merges.
- "git reflog expire branch" can be used in place of "git reflog expire refs/heads/branch".
- "git remote show $remote" lists remote branches one-per-line now.
- "git send-email" can be given revision range instead of files and maildirs on the command line, and automatically runs format-patch to generate patches for the given revision range.
- "git submodule foreach" subcommand allows you to iterate over checked out submodules.
- "git submodule sync" subcommands allows you to update the origin URL recorded in submodule directories from the toplevel .gitmodules file.
- "git svn branch" can create new branches on the other end.
- "gitweb" can use more saner PATH_INFO based URL.

(internal)

- "git hash-object" learned to lie about the path being hashed, so that correct gitattributes processing can be done while hashing contents stored in a temporary file.

- various callers of git-merge-recursive avoid forking it as an external process.

- Git class defined in "Git.pm" can be subclasses a bit more easily.

- We used to link GNU regex library as a compatibility layer for some platforms, but it turns out it is not necessary on most of them.

- Some path handling routines used fixed number of buffers used alternately but depending on the call depth, this arrangement led to hard to track bugs. This issue is being addressed.

Fixes since v1.6.0 ------------------

All of the fixes in v1.6.0.X maintenance series are included in this release, unless otherwise noted.

- Porcelains implemented as shell scripts were utterly confused when you entered to a subdirectory of a work tree from sideways, following a symbolic link (this may need to be backported to older releases later).

- Tracking symbolic links would work better on filesystems whose lstat() returns incorrect st_size value for them.

- "git add" and "git update-index" incorrectly allowed adding S/F when S is a tracked symlink that points at a directory D that has a path F in it (we still need to fix a similar nonsense when S is a submodule and F is a path in it).

- "git am" after stopping at a broken patch lost --whitespace, -C, -p and --3way options given from the command line initially.

- "git diff --stdin" used to take two trees on a line and compared them, but we dropped support for such a use case long time ago. This has been resurrected.

- "git filter-branch" failed to rewrite a tag name with slashes in it.

- "git http-push" did not understand URI scheme other than opaquelocktoken when acquiring a lock from the server (this may need to be backported to older releases later).

- After "git rebase -p" stopped with conflicts while replaying a merge, "git rebase --continue" did not work (may need to be backported to older releases).

- "git revert" records relative to which parent a revert was made when reverting a merge. Together with new documentation that explains issues around reverting a merge and merging from the updated branch later, this hopefully will reduce user confusion (this may need to be backported to older releases later).

- "git rm --cached" used to allow an empty blob that was added earlier to be removed without --force, even when the file in the work tree has since been modified.

- "git push --tags --all $there" failed with generic usage message without telling saying these two options are incompatible.

- "git log --author/--committer" match used to potentially match the timestamp part, exposing internal implementation detail. Also these did not work with --fixed-strings match at all.

- "gitweb" did not mark non-ASCII characters imported from external HTML fragments correctly.

□□□ #2
□ □□□ □□□ 2026-10-31 19:47:21 UTC
□ □□□ □□□ 2026-10-31 19:47:23 UTC