2.8.0

Git 2.8 Release Notes ==========

Backward compatibility note ----------

The rsync:

Updates since v2.7 ----------

UI, Workflows & Features

- It turns out "git clone" over rsync transport has been broken when the source repository has packed references for a long time, and nobody noticed nor complained about it.
- "push" learned that its "--delete" option can be shortened to "-d", just like "branch --delete" and "branch -d" are the same thing.
- "git blame" learned to produce the progress eye-candy when it takes too much time before emitting the first line of the result.
- "git grep" can now be configured (or told from the command line) how many threads to use when searching in the working tree files.
- Some "git notes" operations, e.g. "git log --notes=<note>", should be able to read notes from any tree-ish that is shaped like a notes tree, but the notes infrastructure required that the argument must be a ref under refs/notes/. Loosen it to require a valid ref only when the operation would update the notes (in which case we must have a place to store the updated notes tree, iow, a ref).
- "git grep" by default does not fall back to its "--no-index" behavior outside a directory under Git’s control (otherwise the user may by mistake end up running a huge recursive search); with a new configuration (set in $HOME/.gitconfig—by definition this cannot be set in the config file per project), this safety can be disabled.
- "git pull --rebase" has been extended to allow invoking "rebase -i".
- "git p4" learned to cope with the type of a file getting changed.
- "git format-patch" learned to notice format.outputDirectory configuration variable. This allows "-o <dir>" option to be omitted on the command line if you always use the same directory in your workflow.

- "interpret-trailers" has been taught to optionally update a file in place, instead of always writing the result to the standard output.

- Many commands that read files that are expected to contain text that is generated (or can be edited) by the end user to control their behavior (e.g. "git grep -f <filename>") have been updated to be more tolerant to lines that are terminated with CRLF (they used to treat such a line to contain payload that ends with CR, which is usually not what the users expect).

- "git notes merge" used to limit the source of the merged notes tree to somewhere under refs/notes/ hierarchy, which was too limiting when inventing a workflow to exchange notes with remote repositories using remote-tracking notes trees (located in e.g. refs/remote-notes/ or somesuch).

- "git ls-files" learned a new "--eol" option to help diagnose end-of-line problems.

- "ls-remote" learned an option to show which branch the remote repository advertises as its primary by pointing its HEAD at.

- New http.proxyAuthMethod configuration variable can be used to specify what authentication method to use, as a way to work around proxies that do not give error response expected by libcurl when CURLAUTH_ANY is used. Also, the codepath for proxy authentication has been taught to use credential API to store the authentication material in user’s keyrings.

- Update the untracked cache subsystem and change its primary UI from "git update-index" to "git config".

- There were a few "now I am doing this thing" progress messages in the TCP connection code that can be triggered by setting a verbose option internally in the code, but "git fetch -v" and friends never passed the verbose option down to that codepath.

- Clean/smudge filters defined in a configuration file of lower precedence can now be overridden to be a pass-through no-op by setting the variable to an empty string.

- A new "<branch>^{/!-<pattern>}" notation can be used to name a commit that is reachable from <branch> that does not match the given <pattern>.

- The "user.useConfigOnly" configuration variable can be used to force the user to always set user.email & user.name configuration variables, serving as a reminder for those who work on multiple projects and do not want to put these in their $HOME/.gitconfig.

- "git fetch" and friends that make network connections can now be told to only use ipv4 (or ipv6).

- Some authentication methods do not need username or password, but libcurl needs some hint that it needs to perform authentication. Supplying an empty username and password

string is a valid way to do so, but you can set the http.[<url>.]emptyAuth configuration variable to achieve the same, if you find it cleaner.

- You can now set http.[<url>.]pinnedpubkey to specify the pinned public key when building with recent enough versions of libcURL.

- The configuration system has been taught to phrase where it found a bad configuration variable in a better way in its error messages. "git config" learnt a new "--show-origin" option to indicate where the values come from.

- The "credential-cache" daemon process used to run in whatever directory it happened to start in, but this made umount(2)ing the filesystem that houses the repository harder; now the process chdir()s to the directory that house its own socket on startup.

- When "git submodule update" did not result in fetching the commit object in the submodule that is referenced by the superproject, the command learned to retry another fetch, specifically asking for that commit that may not be connected to the refs it usually fetches.

- "git merge-recursive" learned "--no-renames" option to disable its rename detection logic.

- Across the transition at around Git version 2.0, the user used to get a pretty loud warning when running "git push" without setting push.default configuration variable. We no longer warn because the transition was completed a long time ago.

- README has been renamed to README.md and its contents got tweaked slightly to make it easier on the eyes.

Performance, Internal Implementation, Development Support etc.

- Add a framework to spawn a group of processes in parallel, and use it to run "git fetch --recurse-submodules" in parallel.

- A slight update to the Makefile to mark ".PHONY" targets as such correctly.

- In-core storage of the reverse index for .pack files (which lets you go from a pack offset to an object name) has been streamlined.

- d95138e6 (setup: set env $GIT_WORK_TREE when work tree is set, like $GIT_DIR, 2015-06-26) attempted to work around a glitch in alias handling by overwriting GIT_WORK_TREE environment variable to affect subprocesses when set_git_work_tree() gets called, which resulted in a rather unpleasant regression to "clone" and "init". Try to address the same issue by always restoring the environment and respawning the real underlying command when handling alias.

- The low-level code that is used to create symbolic references has been updated to share more code with the code that deals with normal references.

- strbuf_getline() and friends have been redefined to make it easier to identify which callsite of (new) strbuf_getline_lf() should allow and silently ignore carriage-return at the end of the line to help users on DOSsy systems.

- "git shortlog" used to accumulate various pieces of information regardless of what was asked to be shown in the final output. It has been optimized by noticing what need not to be collected (e.g. there is no need to collect the log messages when showing only the number of changes).

- "git checkout $branch" (and other operations that share the same underlying machinery) has been optimized.

- Automated tests in Travis CI environment has been optimized by persisting runtime statistics of previous "prove" run, executing tests that take longer before other ones; this reduces the total wallclock time.

- Test scripts have been updated to remove assumptions that are not portable between Git for POSIX and Git for Windows, or to skip ones with expectations that are not satisfiable on Git for Windows.

- Some calls to strcpy(3) triggers a false warning from static analyzers that are less intelligent than humans, and reducing the number of these false hits helps us notice real issues. A few calls to strcpy(3) in a couple of programs that are already safe has been rewritten to avoid false warnings.

- The "name_path" API was an attempt to reduce the need to construct the full path out of a series of path components while walking a tree hierarchy, but over time made less efficient because the path needs to be flattened, e.g. to be compared with another path that is already flat. The API has been removed and its users have been rewritten to simplify the overall code complexity.

- Help those who debug http(s) part of the system. (merge 0054045 sp/remote-curl-ssl-strerror later to maint).

- The internal API to interact with "remote.*" configuration variables has been streamlined.

- The ref-filter’s format-parsing code has been refactored, in preparation for "branch --format" and friends.

- Traditionally, the tests that try commands that work on the contents in the working tree were named with "worktree" in their filenames, but with the recent addition of "git worktree" subcommand, whose tests are also named similarly, it has become harder to tell them apart. The traditional tests have been renamed to use "work-tree" instead in an attempt to differentiate them. (merge 5549029 mg/work-tree-tests later to maint).

- Many codepaths forget to check return value from git_config_set(); the function is made to die() to make sure we do not proceed when setting a configuration variable failed. (merge

3d18064 ps/config-error later to maint).

- Handling of errors while writing into our internal asynchronous process has been made more robust, which reduces flakiness in our tests. (merge 43f3afc jk/epipe-in-async later to maint).

- There is a new DEVELOPER knob that enables many compiler warning options in the Makefile.

- The way the test scripts configure the Apache web server has been updated to work also for Apache 2.4 running on RedHat derived distros.

- Out of maintenance gcc on OSX 10.6 fails to compile the code in ’master’; work it around by using clang by default on the platform.

- The "name_path" API was an attempt to reduce the need to construct the full path out of a series of path components while walking a tree hierarchy, but over time made less efficient because the path needs to be flattened, e.g. to be compared with another path that is already flat, in many cases. The API has been removed and its users have been rewritten to simplify the overall code complexity. This incidentally also closes some heap-corruption holes.

- Recent versions of GNU grep is pickier than before to decide if a file is "binary" and refuse to give line-oriented hits when we expect it to, unless explicitly told with "-a" option. As our scripted Porcelains use sane_grep wrapper for line-oriented data, even when the line may contain non-ASCII payload we took from end-user data, use "grep -a" to implement sane_grep wrapper when using an implementation of "grep" that takes the "-a" option.

Also contains various documentation updates and code clean-ups.

Fixes since v2.7 ----------

Unless otherwise noted, all the fixes since v2.7 in the maintenance track are contained in this release (see the maintenance releases’ notes for details).

- An earlier change in 2.5.x-era broke users’ hooks and aliases by exporting GIT_WORK_TREE to point at the root of the working tree, interfering when they tried to use a different working tree without setting GIT_WORK_TREE environment themselves.

- The "exclude_list" structure has the usual "alloc, nr" pair of fields to be used by ALLOC_GROW(), but clear_pattern_list() forgot to reset ’alloc’ to 0 when it cleared ’nr’ to discard the managed array.

- Paths that have been told the index about with "add -N" are not quite yet in the index, but a few commands behaved as if they already are in a harmful way.

- "git send-email" was confused by escaped quotes stored in the alias files saved by "mutt", which has been corrected.

- A few non-portable C construct have been spotted by clang compiler and have been fixed.

- The documentation has been updated to hint the connection between the '--signoff’ option and DCO.

- "git reflog" incorrectly assumed that all objects that used to be at the tip of a ref must be commits, which caused it to segfault.

- The ignore mechanism saw a few regressions around untracked file listing and sparse checkout selection areas in 2.7.0; the change that is responsible for the regression has been reverted.

- Some codepaths used fopen(3) when opening a fixed path in $GIT_DIR (e.g. COMMIT_EDITMSG) that is meant to be left after the command is done. This however did not work well if the repository is set to be shared with core.sharedRepository and the umask of the previous user is tighter. They have been made to work better by calling unlink(2) and retrying after fopen(3) fails with EPERM.

- Asking gitweb for a nonexistent commit left a warning in the server log.

```
Somebody may want to follow this up with an additional test, perhaps?
IIRC, we do test that no Perl warnings are given to the server log,
so this should have been caught if our test coverage were good.
```

- "git rebase", unlike all other callers of "gc --auto", did not ignore the exit code from "gc --auto".

- Many codepaths that run "gc --auto" before exiting kept packfiles mapped and left the file descriptors to them open, which was not friendly to systems that cannot remove files that are open. They now close the packs before doing so.

- A recent optimization to filter-branch in v2.7.0 introduced a regression when --prune-empty filter is used, which has been corrected.

- The description for SANITY prerequisite the test suite uses has been clarified both in the comment and in the implementation.

- "git tag" started listing a tag "foo" as "tags/foo" when a branch named "foo" exists in the same repository; remove this unnecessary disambiguation, which is a regression introduced in v2.7.0.

- The way "git svn" uses auth parameter was broken by Subversion 1.9.0 and later.

- The "split" subcommand of "git subtree" (in contrib/) incorrectly skipped merges when it shouldn’t, which was corrected.

- A few options of "git diff" did not work well when the command was run from a subdirectory.

- The command line completion learned a handful of additional options and command specific syntax.

- dirname() emulation has been added, as Msys2 lacks it.

- The underlying machinery used by "ls-files -o" and other commands has been taught not to create empty submodule ref cache for a directory that is not a submodule. This removes a ton of wasted CPU cycles.

- "git worktree" had a broken code that attempted to auto-fix possible inconsistency that results from end-users moving a worktree to different places without telling Git (the original repository needs to maintain back-pointers to its worktrees, but "mv" run by end-users who are not familiar with that fact will obviously not adjust them), which actually made things worse when triggered.

- The low-level merge machinery has been taught to use CRLF line termination when inserting conflict markers to merged contents that are themselves CRLF line-terminated.

- "git push --force-with-lease" has been taught to report if the push needed to force (or fast-forwarded).

- The emulated "yes" command used in our test scripts has been tweaked not to spend too much time generating unnecessary output that is not used, to help those who test on Windows where it would not stop until it fills the pipe buffer due to lack of SIGPIPE.

- The documentation for "git clean" has been corrected; it mentioned that .git/modules/* are removed by giving two "-f", which has never been the case.

- The vimdiff backend for "git mergetool" has been tweaked to arrange and number buffers in the order that would match the expectation of majority of people who read left to right, then top down and assign buffers 1 2 3 4 "mentally" to local base remote merge windows based on that order.

- "git show ’HEAD:Foo[BAR]Baz’" did not interpret the argument as a rev, i.e. the object named by the pathname with wildcard characters in a tree object. (merge aac4fac nd/dwim-wildcards-as-pathspecs later to maint).

- "git rev-parse --git-common-dir" used in the worktree feature misbehaved when run from a subdirectory. (merge 17f1365 nd/git-common-dir-fix later to maint).

- "git worktree add -B <branchname>" did not work.

- The "v(iew)" subcommand of the interactive "git am -i" command was broken in 2.6.0 timeframe when the command was rewritten in C. (merge 708b8cc jc/am-i-v-fix later to maint).

- "git merge-tree" used to mishandle "both sides added" conflict with its own "create a fake ancestor file that has the common parts of what both sides have added and do a 3-way merge" logic; this has been updated to use the usual "3-way merge with an empty blob as the fake common ancestor file" approach used in the rest of the system. (merge 907681e jk/no-diff-emit-common later to maint).

- The memory ownership rule of fill_textconv() API, which was a bit tricky, has been documented a bit better. (merge a64e6a4 jk/more-comments-on-textconv later to maint).

- Update various codepaths to avoid manually-counted malloc(). (merge 08c95df jk/tighten-alloc later to maint).

- The documentation did not clearly state that the ’simple’ mode is now the default for "git push" when push.default configuration is not set. (merge f6b1fb3 mm/push-simple-doc later to maint).

- Recent versions of GNU grep are pickier when their input contains arbitrary binary data, which some of our tests uses. Rewrite the tests to sidestep the problem. (merge 3b1442d jk/grep-binary-workaround-in-test later to maint).

- A helper function "git submodule" uses since v2.7.0 to list the modules that match the pathspec argument given to its subcommands (e.g. "submodule add <repo> <path>") has been fixed. (merge 2b56bb7 sb/submodule-module-list-fix later to maint).

- "git config section.var value" to set a value in per-repository configuration file failed when it was run outside any repository, but didn’t say the reason correctly. (merge 638fa62 js/config-set-in-non-repository later to maint).

- The code to read the pack data using the offsets stored in the pack idx file has been made more carefully check the validity of the data in the idx. (merge 7465feb jk/pack-idx-corruption-safety later to maint).

- Other minor clean-ups and documentation updates (merge f459823 ak/extract-argv0-last-dir-sep later to maint). (merge 63ca1c0 ak/git-strip-extension-from-dashed-command later to maint). (merge 4867f11 ps/plug-xdl-merge-leak later to maint). (merge 4938686 dt/initial-ref-xn-commit-doc later to maint). (merge 9537f21 ma/update-hooks-sample-typofix later to maint).

□□□ #2
□ □□□ □□□ 2026-10-31 20:05:03 UTC
□ □□□ □□□ 2026-10-31 20:05:08 UTC