2.14.0

Git 2.14 Release Notes ======================

Backward compatibility notes and other notable changes.

- Use of an empty string as a pathspec element that is used for ’everything matches’ is still warned and Git asks users to use a more explicit ’.’ for that instead. The hope is that existing users will not mind this change, and eventually the warning can be turned into a hard error, upgrading the deprecation into removal of this (mis)feature. That is not scheduled to happen in the upcoming release (yet).

- Git now avoids blindly falling back to ".git" when the setup sequence said we are *not* in Git repository. A corner case that happens to work right now may be broken by a call to die("BUG"). We’ve tried hard to locate such cases and fixed them, but there might still be cases that need to be addressed—bug reports are greatly appreciated.

- The experiment to improve the hunk-boundary selection of textual diff output has finished, and the "indent heuristics" has now become the default.

- Git can now be built with PCRE v2 instead of v1 of the PCRE library. Replace USE_LIBPCRE=YesPlease with USE_LIBPCRE2=YesPlease in existing build scripts to build against the new version. As the upstream PCRE maintainer has abandoned v1 maintenance for all but the most critical bug fixes, use of v2 is recommended.

Updates since v2.13 ----------

UI, Workflows & Features

- The colors in which "git status --short --branch" showed the names of the current branch and its remote-tracking branch are now configurable.

- "git clone" learned the "--no-tags" option not to fetch all tags initially, and also set up the tagopt not to follow any tags in subsequent fetches.

- "git archive --format=zip" learned to use zip64 extension when necessary to go beyond the 4GB limit.

- "git reset" learned "--recurse-submodules" option.

- "git diff --submodule=diff" now recurses into nested submodules.

- "git repack" learned to accept the --threads=<n> option and pass it to pack-objects.

- "git send-email" learned to run sendemail-validate hook to inspect and reject a message before sending it out.

- There is no good reason why "git fetch $there $sha1" should fail when the $sha1 names an object at the tip of an advertised ref, even when the other side hasn’t enabled allowTipSHA1InWant.

- The "[includeIf "gitdir:$dir"] path=..." mechanism introduced in 2.13.0 would canonicalize the path of the gitdir being matched, and did not match e.g. "gitdir:~~/work/*" against a repo in "~~/work/main" if "~/work" was a symlink to "/mnt/storage/work". Now we match both the resolved canonical path and what "pwd" would show. The include will happen if either one matches.

- The "indent" heuristics is now the default in "diff". The diff.indentHeuristic configuration variable can be set to "false" for those who do not want it.

- Many commands learned to pay attention to submodule.recurse configuration.

- The convention for a command line is to follow "git cmdname --options" with revisions followed by an optional "--" disambiguator and then finally pathspecs. When "--" is not there, we make sure early ones are all interpretable as revs (and do not look like paths) and later ones are the other way around. A pathspec with "magic" (e.g. ":/p/a/t/h" that matches p/a/t/h from the top-level of the working tree, no matter what subdirectory you are working from) are conservatively judged as "not a path", which required disambiguation more often. The command line parser learned to say "it’s a pathspec" a bit more often when the syntax looks like so.

- Update "perl-compatible regular expression" support to enable JIT and also allow linking with the newer PCRE v2 library.

- "filter-branch" learned a pseudo filter "--setup" that can be used to define common functions/variables that can be used by other filters.

- Using "git add d/i/r" when d/i/r is the top of the working tree of a separate repository would create a gitlink in the index, which would appear as a not-quite-initialized submodule to others. We learned to give warnings when this happens.

- "git status" learned to optionally give how many stash entries there are in its output.

- "git status" has long shown essentially the same message as "git commit"; the message it gives while preparing for the root commit, i.e. "Initial commit", was hard to understand for some new users. Now it says "No commits yet" to stress more on the current status (rather than the commit the user is preparing for, which is more in line with the focus of "git commit").

- "git send-email" now has --batch-size and --relogin-delay options which can be used to overcome limitations on SMTP servers that restrict on how many of e-mails can be sent in

a single session.

- An old message shown in the commit log template was removed, as it has outlived its usefulness.
- "git pull --rebase --recurse-submodules" learns to rebase the branch in the submodules to an updated base.
- "git log" learned -P as a synonym for --perl-regexp, "git grep" already had such a synonym.
- "git log" didn’t understand --regexp-ignore-case when combined with --perl-regexp. This has been fixed.

Performance, Internal Implementation, Development Support etc.

- The default packed-git limit value has been raised on larger platforms to save "git fetch" from a (recoverable) failure while "gc" is running in parallel.
- Code to update the cache-tree has been tightened so that we won’t accidentally write out any 0{40} entry in the tree object.
- Attempt to allow us notice "fishy" situation where we fail to remove the temporary directory used during the test.
- Travis CI gained a task to format the documentation with both AsciiDoc and AsciiDoctor.
- Some platforms have ulong that is smaller than time_t, and our historical use of ulong for timestamp would mean they cannot represent some timestamp that the platform allows. Invent a separate and dedicated timestamp_t (so that we can distinguish timestamps and a vanilla ulongs, which along is already a good move), and then declare uintmax_t is the type to be used as the timestamp_t.
- We can trigger Windows auto-build tester (credits: Dscho & Microsoft) from our existing Travis CI tester now.
- Conversion from uchar[20] to struct object_id continues.
- Simplify parse_pathspec() codepath and stop it from looking at the default in-core index.
- Add perf-test for wildmatch.
- Code from "conversion using external process" codepath has been extracted to a separate sub-process.[ch] module.
- When "git checkout", "git merge", etc. manipulates the in-core index, various pieces of information in the index extensions are discarded from the original state, as it is usually

not the case that they are kept up-to-date and in-sync with the operation on the main index. The untracked cache extension is copied across these operations now, which would speed up "git status" (as long as the cache is properly invalidated).

- The internal implementation of "git grep" has seen some clean-up.

- Update the C style recommendation for notes for translators, as recent versions of gettext tools can work with our style of multi-line comments.

- The implementation of "ref" API around the "packed refs" have been cleaned up, in preparation for further changes.

- The internal logic used in "git blame" has been libified to make it easier to use by cgit.

- Our code often opens a path to an optional file, to work on its contents when we can successfully open it. We can ignore a failure to open if such an optional file does not exist, but we do want to report a failure in opening for other reasons (e.g. we got an I/O error, or the file is there, but we lack the permission to open).

```
The exact errors we need to ignore are ENOENT (obviously) and
ENOTDIR (less obvious).  Instead of repeating comparison of errno
with these two constants, introduce a helper function to do so.
```

- We often try to open a file for reading whose existence is optional, and silently ignore errors from open/fopen; report such errors if they are not due to missing files.

- When an existing repository is used for t/perf testing, we first create bit-for-bit copy of it, which may grab a transient state of the repository and freeze it into the repository used for testing, which then may cause Git operations to fail. Single out "the index being locked" case and forcibly drop the lock from the copy.

- Three instances of the same helper function have been consolidated to one.

- "fast-import" uses a default pack chain depth that is consistent with other parts of the system.

- A new test to show the interaction between the pattern [^a-z] (which matches '/') and a slash in a path has been added. The pattern should not match the slash with "pathmatch", but should with "wildmatch".

- The 'diff-highlight' program (in contrib/) has been restructured for easier reuse by an external project 'diff-so-fancy'.

- A common pattern to free a piece of memory and assign NULL to the pointer that used to point at it has been replaced with a new FREE_AND_NULL() macro.

- Traditionally, the default die() routine had a code to prevent it from getting called multiple times, which interacted badly when a threaded program used it (one downside is that the real error may be hidden and instead the only error message given to the user may end up being "die recursion detected", which is not very useful).

- Introduce a "repository" object to eventually make it easier to work in multiple repositories (the primary focus is to work with the superproject and its submodules) in a single process.

- Optimize "what are the object names already taken in an alternate object database?" query that is used to derive the length of prefix an object name is uniquely abbreviated to.

- The hashmap API has been updated so that data to customize the behaviour of the comparison function can be specified at the time a hashmap is initialized.

- The "collision detecting" SHA-1 implementation shipped with 2.13 is now integrated into git.git as a submodule (the first submodule to ship with git.git). Clone git.git with --recurse-submodules to get it. For now a non-submodule copy of the same code is also shipped as part of the tree.

- A recent update made it easier to use "-fsanitize=" option while compiling but supported only one sanitize option. Allow more than one to be combined, joined with a comma, like "make SANITIZE=foo,bar".

- Use "p4 -G" to make "p4 changes" output more Python-friendly to parse.

- We started using "%" PRItime, imitating "%" PRIuMAX and friends, as a way to format the internal timestamp value, but this does not play well with gettext(1) i18n framework, and causes "make pot" that is run by the l10n coordinator to create a broken po/git.pot file. This is a possible workaround for that problem.

- It turns out that Cygwin also needs the fopen() wrapper that returns failure when a directory is opened for reading.

Also contains various documentation updates and code clean-ups.

Fixes since v2.13 ----------

Unless otherwise noted, all the fixes since v2.13 in the maintenance track are contained in this release (see the maintenance releases’ notes for details).

- "git gc" did not interact well with "git worktree"-managed per-worktree refs.

- "git cherry-pick" and other uses of the sequencer machinery mishandled a trailer block whose last line is an incomplete line. This has been fixed so that an additional sign-off etc. are added after completing the existing incomplete line.

- The codepath in "git am" that is used when running "git rebase" leaked memory held for the log message of the commits being rebased.

- "git clone --config var=val" is a way to populate the per-repository configuration file of the new repository, but it did not work well when val is an empty string. This has been fixed.

- Setting "log.decorate=false" in the configuration file did not take effect in v2.13, which has been corrected.

- A few codepaths in "checkout" and "am" working on an unborn branch tried to access an uninitialized piece of memory.

- The Web interface to gmane news archive is long gone, even though the articles are still accessible via NTTP. Replace the links with ones to public-inbox.org. Because their message identification is based on the actual message-id, it is likely that it will be easier to migrate away from it if/when necessary.

- The receive-pack program now makes sure that the push certificate records the same set of push options used for pushing.

- Tests have been updated to pass under GETTEXT_POISON (a mechanism to ensure that output strings that should not be translated are not translated by mistake), and TravisCI is told to run them.

- "git checkout --recurse-submodules" did not quite work with a submodule that itself has submodules.

- "pack-objects" can stream a slice of an existing packfile out when the pack bitmap can tell that the reachable objects are all needed in the output, without inspecting individual objects. This strategy however would not work well when "--local" and other options are in use, and need to be disabled.

- Fix memory leaks pointed out by Coverity (and people).

- "git read-tree -m" (no tree-ish) gave a nonsense suggestion "use --empty if you want to clear the index". With "-m", such a request will still fail anyway, as you'd need to name at least one tree-ish to be merged.

- Make sure our tests would pass when the sources are checked out with "platform native" line ending convention by default on Windows. Some "text" files out tests use and the test scripts themselves that are meant to be run with /bin/sh, ought to be checked out with eol=LF even on Windows.

- Introduce the BUG() macro to improve die("BUG: ...").

- Clarify documentation for include.path and includeIf.<condition>.path configuration variables.

- Git sometimes gives an advice in a rhetorical question that does not require an answer, which can confuse new users and non native speakers. Attempt to rephrase them.

- A few http:// links that are redirected to https:// in the documentation have been updated to https:// links.

- "git for-each-ref --format=..." with %(HEAD) in the format used to resolve the HEAD symref as many times as it had processed refs, which was wasteful, and "git branch" shared the same problem.

- Regression fix to topic recently merged to 'master'.

- The shell completion script (in contrib/) learned "git stash" has a new "push" subcommand.

- "git interpret-trailers", when used as GIT_EDITOR for "git commit -v", looked for and appended to a trailer block at the very end, i.e. at the end of the "diff" output. The command has been corrected to pay attention to the cut-mark line "commit -v" adds to the buffer---the real trailer block should appear just before it.

- A test allowed both "git push" and "git receive-pack" on the other end write their traces into the same file. This is OK on platforms that allows atomically appending to a file opened with O_APPEND, but on other platforms led to a mangled output, causing intermittent test failures. This has been fixed by disabling traces from "receive-pack" in the test.

- Tag objects, which are not reachable from any ref, that point at missing objects were mishandled by "git gc" and friends (they should silently be ignored instead)

- "git describe --contains" penalized light-weight tags so much that they were almost never considered. Instead, give them about the same chance to be considered as an annotated tag that is the same age as the underlying commit would.

- The "run-command" API implementation has been made more robust against dead-locking in a threaded environment.

- A recent update to t5545-push-options.sh started skipping all the tests in the script when a web server testing is disabled or unavailable, not just the ones that require a web server. Non HTTP tests have been salvaged to always run in this script.

- "git send-email" now uses Net::SMTP::SSL, which is obsolete, only when needed. Recent versions of Net::SMTP can do TLS natively.

- "foobarbaz" in "git fetch foobarbaz", even though there is no slashes in it, cannot be a nickname for a remote on Windows, as that is likely to be a pathname on a local filesystem.

- "git clean -d" used to clean directories that has ignored files, even though the command should not lose ignored ones without "-x". "git status --ignored" did not list ignored and untracked files without "-uall". These have been corrected.

- The result from "git diff" that compares two blobs, e.g. "git diff $commit1:$path $commit2:$path", used to be shown with the full object name as given on the command line, but it is more natural to use the $path in the output and use it to look up .gitattributes.

- The "collision detecting" SHA-1 implementation shipped with 2.13 was quite broken on some big-endian platforms and/or platforms that do not like unaligned fetches. Update to the upstream code which has already fixed these issues.

- "git am -h" triggered a BUG().

- The interaction of "url.*.insteadOf" and custom URL scheme’s whitelisting is now documented better.

- The timestamp of the index file is now taken after the file is closed, to help Windows, on which a stale timestamp is reported by fstat() on a file that is opened for writing and data was written but not yet closed.

- "git pull --rebase --autostash" didn’t auto-stash when the local history fast-forwards to the upstream.

- A flaky test has been corrected.

- "git $cmd -h" for builtin commands calls the implementation of the command (i.e. cmd_$cmd() function) without doing any repository set-up, and the commands that expect RUN_SETUP is done by the Git potty needs to be prepared to show the help text without barfing. (merge d691551192 jk/consistent-h later to maint).

- Help contributors that visit us at GitHub.

- "git stash push <pathspec>" did not work from a subdirectory at all. Bugfix for a topic in v2.13

- As there is no portable way to pass timezone information to strftime, some output format from "git log" and friends are impossible to produce. Teach our own strbuf_addftime to replace %z and %Z with caller-supplied values to help working around this. (merge 6eced3ec5e rs/strbuf-addftime-zZ later to maint).

- "git mergetool" learned to work around a wrapper MacOS X adds around underlying meld.

- An example in documentation that does not work in multi worktree configuration has been corrected.

- The pretty-format specifiers like ’%h’, ’%t’, etc. had an optimization that no longer works correctly. In preparation/hope of getting it correctly implemented, first discard the optimization that is broken.

- The code to pick up and execute command alias definition from the configuration used to switch to the top of the working tree and then come back when the expanded alias was executed, which was unnecessarily complex. Attempt to simplify the logic by using the early-config mechanism that does not chdir around.

- Fix configuration codepath to pay proper attention to commondir that is used in multi-worktree situation, and isolate config API into its own header file. (merge dc8441fdb4 bw/config-h later to maint).

- "git add -p" were updated in 2.12 timeframe to cope with custom core.commentchar but the implementation was buggy and a metacharacter like $ and * did not work.

- A recent regression in "git rebase -i" has been fixed and tests that would have caught it and others have been added.

- An unaligned 32-bit access in pack-bitmap code has been corrected.

- Tighten error checks for invalid "git apply" input.

- The split index code did not honor core.sharedRepository setting correctly.

- The Makefile rule in contrib/subtree for building documentation learned to honour USE_ASCIIDOCTOR just like the main documentation set does.

- Code clean-up to fix possible buffer over-reading.

- A few tests that tried to verify the contents of push certificates did not use ’git rev-parse’ to formulate the line to look for in the certificate correctly.

- Update the character width tables.

- After "git branch --move" of the currently checked out branch, the code to walk the reflog of HEAD via "log -g" and friends incorrectly stopped at the reflog entry that records the renaming of the branch.

- The rewrite of "git branch --list" using for-each-ref’s internals that happened in v2.13 regressed its handling of color.branch.local; this has been fixed.

- The build procedure has been improved to allow building and testing Git with address sanitizer more easily. (merge 425ca6710b jk/build-with-asan later to maint).

- On Cygwin, similar to Windows, "git push //server/share/repository" ought to mean a repository on a network share that can be accessed locally, but this did not work correctly

due to stripping the double slashes at the beginning.

- The progress meter did not give a useful output when we haven’t had 0.5 seconds to measure the throughput during the interval. Instead show the overall throughput rate at the end, which is a much more useful number.

- Code clean-up, that makes us in sync with Debian by one patch.

- We run an early part of "git gc" that deals with refs before daemonising (and not under lock) even when running a background auto-gc, which caused multiple gc processes attempting to run the early part at the same time. This is now prevented by running the early part also under the GC lock.

- A recent update broke an alias that contained an uppercase letter.

- Other minor doc, test and build updates and code cleanups. (merge 5053313562 rs/urlmatch-cleanup later to maint). (merge 42c78a216e rs/use-div-round-up later to maint). (merge 5e8d2729ae rs/wt-status-cleanup later to maint). (merge bc9b7e207f as/diff-options-grammofix later to maint). (merge ac05222b31 ah/patch-id-doc later to maint).

□□□ #2
□ □□□ □□□ 2026-10-31 19:54:32 UTC
□ □□□ □□□ 2026-10-31 19:54:34 UTC