2.2.0

Git v2.2 Release Notes ======================

Updates since v2.1 ------------------

Ports

- Building on older MacOS X systems automatically sets the necessary NO_APPLE_COMMON_CRYPTO build-time option.
- Building with NO_PTHREADS has been resurrected.
- Compilation options have been updated a bit to better support the z/OS port.

UI, Workflows & Features

- "git archive" learned to filter what gets archived with a pathspec.
- "git config --edit --global" starts from a skeletal per-user configuration file contents, instead of a total blank, when the user does not already have any global config. This immediately reduces the need to later ask "Have you forgotten to set core.user?", and we can add more to the template as we gain more experience.
- "git stash list -p" used to be almost always a no-op because each stash entry is represented as a merge commit. It learned to show the difference between the base commit version and the working tree version, which is in line with what "git stash show" gives.
- Sometimes users want to report a bug they experience on their repository, but they are not at liberty to share the contents of the repository. "fast-export" was taught an "--anonymize" option to replace blob contents, names of people, paths and log messages with bland and simple strings to help them.
- "git difftool" learned an option to stop feeding paths to the diff backend when it exits with a non-zero status.
- "git grep" learned to paint (or not paint) partial matches on context lines when showing "grep -C<num>" output in color.
- "log --date=iso" uses a slight variant of the ISO 8601 format that is more human readable. A new "--date=iso-strict" option gives datetime output that conforms more strictly.

- The logic "git prune" uses is more resilient against various corner cases.

- A broken reimplementation of Git could write an invalid index that records both stage #0 and higher-stage entries for the same path. We now notice and reject such an index, as there is no sensible fallback (we do not know if the broken tool wanted to resolve and forgot to remove the higher-stage entries, or if it wanted to unresolve and forgot to remove the stage #0 entry).

- The temporary files "git mergetool" uses are renamed to avoid too many dots in them (e.g. a temporary file for "hello.c" used to be named e.g. "hello.BASE.4321.c" but now uses underscore instead, e.g. "hello_BASE_4321.c", to allow us to have multiple variants).

- The temporary files "git mergetool" uses can be placed in a newly created temporary directory, instead of the current directory, by setting the mergetool.writeToTemp configuration variable.

- "git mergetool" understands "--tool bc" now, as version 4 of BeyondCompare can be driven the same way as its version 3 and it feels awkward to say "--tool bc3" to run version 4.

- The "pre-receive" and "post-receive" hooks are no longer required to consume their input fully (not following this requirement used to result in intermittent errors in "git push").

- The pretty-format specifier "%d", which expands to " (tagname)" for a tagged commit, gained a cousin "%D" that just gives the "tagname" without frills.

- "git push" learned "--signed" push, that allows a push (i.e. request to update the refs on the other side to point at a new history, together with the transmission of necessary objects) to be signed, so that it can be verified and audited, using the GPG signature of the person who pushed, that the tips of branches at a public repository really point the commits the pusher wanted to, without having to "trust" the server.

- "git interpret-trailers" is a new filter to programmatically edit the tail end of the commit log messages, e.g. "Signed-off-by:".

- "git help everyday" shows the "Everyday Git in 20 commands or so" document, whose contents have been updated to match more modern Git practice.

- On the "git svn" front, work progresses to reduce memory consumption and to improve handling of mergeinfo.

Performance, Internal Implementation, Development Support etc.

- The API to manipulate the "refs" has been restructured to make it more transactional, with the eventual goal to allow all-or-none atomic updates and migrating the storage to something other than the traditional filesystem based one (e.g. databases).

- The lockfile API and its users have been cleaned up.
- We no longer attempt to keep track of individual dependencies to the header files in the build procedure, relying instead on automated dependency generation support from modern compilers.
- In tests, we have been using NOT_{MINGW,CYGWIN} test prerequisites long before negated prerequisites e.g. !MINGW were invented. The former has been converted to the latter to avoid confusion.
- Optimized looking up a remote’s configuration in a repository with very many remotes defined.
- There are cases where you lock and open to write a file, close it to show the updated contents to an external processes, and then have to update the file again while still holding the lock; now the lockfile API has support for such an access pattern.
- The API to allocate the structure to keep track of commit decoration has been updated to make it less cumbersome to use.
- An in-core caching layer to let us avoid reading the same configuration files several times has been added. A few commands have been converted to use this subsystem.
- Various code paths have been cleaned up and simplified by using the "strbuf", "starts_with()", and "skip_prefix()" APIs more.
- A few codepaths that died when large blobs that would not fit in core are involved in their operation have been taught to punt instead, by e.g. marking a too-large blob as not to be diffed.
- A few more code paths in "commit" and "checkout" have been taught to repopulate the cache-tree in the index, to help speed up later "write-tree" (used in "commit") and "diff-index --cached" (used in "status").
- A common programming mistake to assign the same short option name to two separate options is detected by the parse_options() API to help developers.
- The code path to write out the packed-refs file has been optimized, which especially matters in a repository with a large number of refs.
- The check to see if a ref $F can be created by making sure no existing ref has $F/ as its prefix has been optimized, which especially matters in a repository with a large number of existing refs.
- "git fsck" was taught to check the contents of tag objects a bit more.
- "git hash-object" was taught a "--literally" option to help debugging.

- When running a required clean filter, we do not have to mmap the original before feeding the filter. Instead, stream the file contents directly to the filter and process its output.

- The scripts in the test suite can be run with the "-x" option to show a shell-trace of each command they run.

- The "run-command" API learned to manage the argv and environment arrays for child process, alleviating the need for the callers to allocate and deallocate them.

- Some people use AsciiDoctor, instead of AsciiDoc, to format our documentation set; the documentation has been adjusted to be usable by both, as AsciiDoctor is pickier than AsciiDoc about its input mark-up.

Also contains various documentation updates and code clean-ups.

Fixes since v2.1 ----------

Unless otherwise noted, all the fixes since v2.1 in the maintenance track are contained in this release (see the maintenance releases' notes for details).

- "git log --pretty/format=" with an empty format string did not mean the more obvious "No output whatsoever" but "Use default format", which was counterintuitive.

- "git -c section.var command" and "git -c section.var= command" should pass the configuration value differently (the former should be a boolean true, the latter should be an empty string).

- Applying a patch not generated by Git in a subdirectory used to check for whitespace breakage using the attributes of incorrect paths. Also whitespace checks were performed even for paths excluded via the "git apply --exclude=<path>" mechanism.

- "git bundle create" with a date-range specification was meant to exclude tags outside the range, but it didn't.

- "git add x" where x used to be a directory and is now a symbolic link to a directory misbehaved.

- The prompt script checked the $GIT_DIR/ref/stash file to see if there is a stash, which was a no-no.

- Pack-protocol documentation had a minor typo.

- "git checkout -m" did not switch to another branch while carrying the local changes forward when a path was deleted from the index.

- "git daemon" (with NO_IPV6 build configuration) used to incorrectly use the hostname even when gethostbyname() reported that the given hostname is not found. (merge

107efbe rs/daemon-fixes later to maint).

- With sufficiently long refnames, "git fast-import" could have overflowed an on-stack buffer.

- After "pack-refs --prune" packed refs at the top-level, it failed to prune them.

- Progress output from "git gc --auto" was visible in "git fetch -q".

- We used to pass -1000 to poll(2), expecting it to also mean "no timeout", which should be spelled as -1.

- "git rebase" documentation was unclear that it is required to specify on what <upstream> the rebase is to be done when telling it to first check out <branch>. (merge 95c6826 so/rebase-doc later to maint).

- "git push" over HTTP transport had an artificial limit on the number of refs that can be pushed, imposed by the command line length. (merge 26be19b jk/send-pack-many-refspecs later to maint).

- When receiving an invalid pack stream that records the same object twice, multiple threads got confused due to a race. (merge ab791dd jk/index-pack-threading-races later to maint).

- An attempt to remove the entire tree in the "git fast-import" input stream caused it to misbehave. (merge 2668d69 mb/fast-import-delete-root later to maint).

- Reachability check (used in "git prune" and friends) did not add a detached HEAD as a starting point to traverse objects still in use. (merge c40fdd0 mk/reachable-protect-detached-head later to maint).

- "git config --add section.var val" when section.var already has an empty-string value used to lose the empty-string value. (merge c1063be ta/config-add-to-empty-or-true-fix later to maint).

- "git fsck" failed to report that it found corrupt objects via its exit status in some cases. (merge 30d1038 jk/fsck-exit-code-fix later to maint).

- Use of the "--verbose" option used to break "git branch --merged". (merge 12994dd jk/maint-branch-verbose-merged later to maint).

- Some MUAs mangle a line in a message that begins with "From " to ">From " when writing to a mailbox file, and feeding such an input to "git am" used to lose such a line. (merge 85de86a jk/mbox-from-line later to maint).

- "rev-parse --verify --quiet $name" is meant to quietly exit with a non-zero status when $name is not a valid object name, but still gave error messages in some cases.

- A handful of C source files have been updated to include "git-compat-util.h" as the first thing, to conform better to our coding guidelines. (merge 1c4b660 da/include-compat-util-first-in-c later to maint).

- The t7004 test, which tried to run Git with small stack space, has been updated to use a bit larger stack to avoid false breakage on some platforms. (merge b9a1907 sk/tag-contains-wo-recursion later to maint).

- A few documentation pages had example sections marked up not quite correctly, which passed AsciiDoc but failed with AsciiDoctor. (merge c30c43c bc/asciidoc-pretty-formats-fix later to maint). (merge f8a48af bc/asciidoc later to maint).

- "gitweb" used deprecated CGI::startfrom, which was removed from CGI.pm as of 4.04; use CGI::start_from instead. (merge 4750f4b rm/gitweb-start-form later to maint).

- Newer versions of ’meld’ break the auto-detection we use to see if they are new enough to support the `--output` option. (merge b12d045 da/mergetool-meld later to maint).

- "git pack-objects" forgot to disable the codepath to generate the object reachability bitmap when it needs to split the resulting pack. (merge 2113471 jk/pack-objects-no-bitmap-when-splitting later to maint).

- The code to use cache-tree trusted the on-disk data too much and fell into an infinite loop upon seeing an incorrectly recorded index file. (merge 729dbbd jk/cache-tree-protect-from-broken-libgit2 later to maint).

- "git fetch" into a repository where branch B was deleted earlier, back when it had reflog enabled, and then branch B/C is fetched into it without reflog enabled, which is arguably an unlikely corner case, unnecessarily failed. (merge aae828b jk/fetch-reflog-df-conflict later to maint).

- "git log --first-parent -L..." used to crash. (merge a8787c5 tm/line-log-first-parent later to maint).

□□□ #2
□ □□□ □□□ 2026-10-31 19:56:00 UTC
□ □□□ □□□ 2026-10-31 19:56:01 UTC