2.11.0

Git 2.11 Release Notes ======================

Backward compatibility notes.

- An empty string used as a pathspec element has always meant ’everything matches’, but it is too easy to write a script that finds a path to remove in $path and run ’git rm "$paht"’ by mistake (when the user meant to give "$path"), which ends up removing everything. This release starts warning about the use of an empty string that is used for ’everything matches’ and 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.
```

- The historical argument order "git merge <msg> HEAD <commit>..." has been deprecated for quite some time, and will be removed in the next release (not this one).

- The default abbreviation length, which has historically been 7, now scales as the repository grows, using the approximate number of objects in the repository and a bit of math around the birthday paradox. The logic suggests to use 12 hexdigits for the Linux kernel, and 9 to 10 for Git itself.

Updates since v2.10 ----------

UI, Workflows & Features

- Comes with new version of git-gui, now at its 0.21.0 tag.

- "git format-patch --cover-letter HEAD^" to format a single patch with a separate cover letter now numbers the output as [PATCH 0/1] and [PATCH 1/1] by default.

- An incoming "git push" that attempts to push too many bytes can now be rejected by setting a new configuration variable at the receiving end.

- "git nosuchcommand --help" said "No manual entry for gitnosuchcommand", which was not intuitive, given that "git nosuchcommand" said "git: ’nosuchcommand’ is not a git command".

- "git clone --recurse-submodules --reference $path $URL" is a way to reduce network transfer cost by borrowing objects in an existing $path repository when cloning the

superproject from $URL; it learned to also peek into $path for presence of corresponding repositories of submodules and borrow objects from there when able.

- The "git diff --submodule={short,log}" mechanism has been enhanced to allow "--submodule=diff" to show the patch between the submodule commits bound to the superproject.

- Even though "git hash-objects", which is a tool to take an on-filesystem data stream and put it into the Git object store, can perform "outside-world-to-Git" conversions (e.g. end-of-line conversions and application of the clean-filter), and it has had this feature on by default from very early days, its reverse operation "git cat-file", which takes an object from the Git object store and externalizes it for consumption by the outside world, lacked an equivalent mechanism to run the "Git-to-outside-world" conversion. The command learned the "--filters" option to do so.

- Output from "git diff" can be made easier to read by intelligently selecting which lines are common and which lines are added/deleted when the lines before and after the changed section are the same. A command line option (--indent-heuristic) and a configuration variable (diff.indentHeuristic) are added to help with the experiment to find good heuristics.

- In some projects, it is common to use "[RFC PATCH]" as the subject prefix for a patch meant for discussion rather than application. A new format-patch option "--rfc" is a short-hand for "--subject-prefix=RFC PATCH" to help the participants of such projects.

- "git add --chmod={+,-}x <pathspec>" only changed the executable bit for paths that are either new or modified. This has been corrected to change the executable bit for all paths that match the given pathspec.

- When "git format-patch --stdout" output is placed as an in-body header and it uses RFC2822 header folding, "git am" fails to put the header line back into a single logical line. The underlying "git mailinfo" was taught to handle this properly.

- "gitweb" can spawn "highlight" to show blob contents with (programming) language-specific syntax highlighting, but only when the language is known. "highlight" can however be told to guess the language itself by giving it "--force" option, which has been enabled.

- "git gui" l10n to Portuguese.

- When given an abbreviated object name that is not (or more realistically, "no longer") unique, we gave a fatal error "ambiguous argument". This error is now accompanied by a hint that lists the objects beginning with the given prefix. During the course of development of this new feature, numerous minor bugs were uncovered and corrected, the most notable one of which is that we gave "short SHA1 xxxx is ambiguous." twice without good reason.

- "git log rev^..rev" is an often-used revision range specification to show what was done on a side branch merged at rev. This has gained a short-hand "rev^-1". In general "rev^-$n" is the same as "rev$n rev", i.e. what has happened on other branches while the history leading to nth parent was looking the other way.

- In recent versions of cURL, GSSAPI credential delegation is disabled by default due to CVE-2011-2192; introduce a http.delegation configuration variable to selectively allow enabling this. (merge 26a7b23429 ps/http-gssapi-cred-delegation later to maint).

- "git mergetool" learned to honor "-O<orderfile>" to control the order of paths to present to the end user.

- "git diff/log --ws-error-highlight=<kind>" lacked the corresponding configuration variable (diff.wsErrorHighlight) to set it by default.

- "git ls-files" learned the "--recurse-submodules" option to get a listing of tracked files across submodules (i.e. this only works with the "--cached" option, not for listing untracked or ignored files). This would be a useful tool to sit on the upstream side of a pipe that is read with xargs to work on all working tree files from the top-level superproject.

- A new credential helper that talks via "libsecret" with implementations of XDG Secret Service API has been added to contrib/credential/.

- The GPG verification status shown by the "%G?" pretty format specifier was not rich enough to differentiate a signature made by an expired key, a signature made by a revoked key, etc. New output letters have been assigned to express them.

- In addition to purely abbreviated commit object names, "gitweb" learned to turn "git describe" output (e.g. v2.9.3-599-g2376d31787) into clickable links in its output.

- "git commit" created an empty commit when invoked with an index consisting solely of intend-to-add paths (added with "git add -N"). It now requires the "--allow-empty" option to create such a commit. The same logic prevented "git status" from showing such paths as "new files" in the "Changes not staged for commit" section.

- The smudge/clean filter API spawns an external process to filter the contents of each path that has a filter defined. A new type of "process" filter API has been added to allow the first request to run the filter for a path to spawn a single process, and all filtering is served by this single process for multiple paths, reducing the process creation overhead.

- The user always has to say "stash@{$N}" when naming a single element in the default location of the stash, i.e. reflogs in refs/stash. The "git stash" command learned to accept "git stash apply 4" as a short-hand for "git stash apply stash@{4}".

Performance, Internal Implementation, Development Support etc.

- The delta-base-cache mechanism has been a key to the performance in a repository with a tightly packed packfile, but it did not scale well even with a larger value of core.deltaBaseCacheLimit.

- Enhance "git status --porcelain" output by collecting more data on the state of the index and the working tree files, which may further be used to teach git-prompt (in contrib/) to make fewer calls to git.

- Extract a small helper out of the function that reads the authors script file "git am" internally uses. (merge a77598e jc/am-read-author-file later to maint).

- Lift calls to exit(2) and die() higher in the callchain in sequencer.c files so that more helper functions in it can be used by callers that want to handle error conditions themselves.

- "git am" has been taught to make an internal call to "git apply"'s innards without spawning the latter as a separate process.

- The ref-store abstraction was introduced to the refs API so that we can plug in different backends to store references.

- The "unsigned char sha1[20]" to "struct object_id" conversion continues. Notable changes in this round includes that ce→sha1, i.e. the object name recorded in the cache_entry, turns into an object_id.

- JGit can show a fake ref "capabilities^{}" to "git fetch" when it does not advertise any refs, but "git fetch" was not prepared to see such an advertisement. When the other side disconnects without giving any ref advertisement, we used to say "there may not be a repository at that URL", but we may have seen other advertisements like "shallow" and ".have" in which case we definitely know that a repository is there. The code to detect this case has also been updated.

- Some codepaths in "git pack-objects" were not ready to use an existing pack bitmap; now they are and as a result they have become faster.

- The codepath in "git fsck" to detect malformed tree objects has been updated not to die but keep going after detecting them.

- We call "qsort(array, nelem, sizeof(array[0]), fn)", and most of the time third parameter is redundant. A new QSORT() macro lets us omit it.

- "git pack-objects" in a repository with many packfiles used to spend a lot of time looking for/at objects in them; the accesses to the packfiles are now optimized by checking the most-recently-used packfile first. (merge c9af708b1a jk/pack-objects-optim-mru later to maint).

- Codepaths involved in interacting alternate object stores have been cleaned up.

- In order for the receiving end of "git push" to inspect the received history and decide to reject the push, the objects sent from the sending end need to be made available to the hook and the mechanism for the connectivity check, and this was done traditionally by storing the objects in the receiving repository and letting "git gc" expire them. Instead, store the newly received objects in a temporary area, and make them available by reusing the alternate object store mechanism to them only while we decide if we accept the check, and once we decide, either migrate them to the repository or purge them immediately.

- The require_clean_work_tree() helper was recreated in C when "git pull" was rewritten from shell; the helper is now made available to other callers in preparation for upcoming "rebase -i" work.

- "git upload-pack" had its code cleaned-up and performance improved by reducing use of timestamp-ordered commit-list, which was replaced with a priority queue.

- "git diff --no-index" codepath has been updated not to try to peek into a .git/ directory that happens to be under the current directory, when we know we are operating outside any repository.

- Update of the sequencer codebase to make it reusable to reimplement "rebase -i" continues.

- Git generally does not explicitly close file descriptors that were open in the parent process when spawning a child process, but most of the time the child does not want to access them. As Windows does not allow removing or renaming a file that has a file descriptor open, a slow-to-exit child can even break the parent process by holding onto them. Use O_CLOEXEC flag to open files in various codepaths.

- Update "interpret-trailers" machinery and teach it that people in the real world write all sorts of cruft in the "trailer" that was originally designed to have the neat-o "Mail-Header: like thing" and nothing else.

Also contains various documentation updates and code clean-ups.

Fixes since v2.10 ----------

Unless otherwise noted, all the fixes since v2.9 in the maintenance track are contained in this release (see the maintenance releases' notes for details).

- Clarify various ways to specify the "revision ranges" in the documentation.

- "diff-highlight" script (in contrib/) learned to work better with "git log -p --graph" output.

- The test framework left the number of tests and success/failure count in the t/test-results directory, keyed by the name of the test script plus the process ID. The latter however turned out not to serve any useful purpose. The process ID part of the filename has been

removed.

- Having a submodule whose ".git" repository is somehow corrupt caused a few commands that recurse into submodules to loop forever.

- "git symbolic-ref -d HEAD" happily removes the symbolic ref, but the resulting repository becomes an invalid one. Teach the command to forbid removal of HEAD.

- A test spawned a short-lived background process, which sometimes prevented the test directory from getting removed at the end of the script on some platforms.

- Update a few tests that used to use GIT_CURL_VERBOSE to use the newer GIT_TRACE_CURL.

- "git pack-objects --include-tag" was taught that when we know that we are sending an object C, we want a tag B that directly points at C but also a tag A that points at the tag B. We used to miss the intermediate tag B in some cases.

- Update Japanese translation for "git-gui".

- "git fetch http::/site/path" did not die correctly and segfaulted instead.

- "git commit-tree" stopped reading commit.gpgsign configuration variable that was meant for Porcelain "git commit" in Git 2.9; we forgot to update "git gui" to look at the configuration to match this change.

- "git add --chmod={+,-}x" added recently lacked documentation, which has been corrected.

- "git log --cherry-pick" used to include merge commits as candidates to be matched up with other commits, resulting a lot of wasted time. The patch-id generation logic has been updated to ignore merges and avoid the wastage.

- The http transport (with curl-multi option, which is the default these days) failed to remove curl-easy handle from a curlm session, which led to unnecessary API failures.

- There were numerous corner cases in which the configuration files are read and used or not read at all depending on the directory a Git command was run, leading to inconsistent behaviour. The code to set-up repository access at the beginning of a Git process has been updated to fix them. (merge 4d0efa1 jk/setup-sequence-update later to maint).

- "git diff -W" output needs to extend the context backward to include the header line of the current function and also forward to include the body of the entire current function up to the header line of the next one. This process may have to merge two adjacent hunks, but the code forgot to do so in some cases.

- Performance tests done via "t/perf" did not use the right build configuration if the user relied on autoconf generated configuration.

- "git format-patch --base=..." feature that was recently added showed the base commit information after the "-- " e-mail signature line, which turned out to be inconvenient. The base information has been moved above the signature line.

- More i18n.

- Even when "git pull --rebase=preserve" (and the underlying "git rebase --preserve") can complete without creating any new commits (i.e. fast-forwards), it still insisted on having usable ident information (read: user.email is set correctly), which was less than nice. As the underlying commands used inside "git rebase" would fail with a more meaningful error message and advice text when the bogus ident matters, this extra check was removed.

- "git gc --aggressive" used to limit the delta-chain length to 250, which is way too deep for gaining additional space savings and is detrimental for runtime performance. The limit has been reduced to 50.

- Documentation for individual configuration variables to control use of color (like `color.grep`) said that their default value is 'false', instead of saying their default is taken from `color.ui`. When we updated the default value for color.ui from 'false' to 'auto' quite a while ago, all of them broke. This has been corrected.

- The pretty-format specifier "%C(auto)" used by the "log" family of commands to enable coloring of the output is taught to also issue a color-reset sequence to the output.

- A shell script example in check-ref-format documentation has been fixed.

- "git checkout <word>" does not follow the usual disambiguation rules when the <word> can be both a rev and a path, to allow checking out a branch 'foo' in a project that happens to have a file 'foo' in the working tree without having to disambiguate. This was poorly documented and the check was incorrect when the command was run from a subdirectory.

- Some codepaths in "git diff" used regexec(3) on a buffer that was mmap(2)ed, which may not have a terminating NUL, leading to a read beyond the end of the mapped region. This was fixed by introducing a regexec_buf() helper that takes a <ptr,len> pair with REG_STARTEND extension.

- The procedure to build Git on Mac OS X for Travis CI hardcoded the internal directory structure we assumed HomeBrew uses, which was a no-no. The procedure has been updated to ask HomeBrew things we need to know to fix this.

- When "git rebase -i" is given a broken instruction, it told the user to fix it with "--edit-todo", but didn't say what the step after that was (i.e. "--continue").

- Documentation around tools to import from CVS was fairly outdated.
- "git clone --recurse-submodules" lost the progress eye-candy in a recent update, which has been corrected.
- A low-level function verify_packfile() was meant to show errors that were detected without dying itself, but under some conditions it didn’t and died instead, which has been fixed.
- When "git fetch" tries to find where the history of the repository it runs in has diverged from what the other side has, it has a mechanism to avoid digging too deep into irrelevant side branches. This however did not work well over the "smart-http" transport due to a design bug, which has been fixed.
- In the codepath that comes up with the hostname to be used in an e-mail when the user didn’t tell us, we looked at the ai_canonname field in struct addrinfo without making sure it is not NULL first.
- "git worktree", even though it used the default_abbrev setting that ought to be affected by the core.abbrev configuration variable, ignored the variable setting. The command has been taught to read the default set of configuration variables to correct this.
- "git init" tried to record core.worktree in the repository’s ’config’ file when the GIT_WORK_TREE environment variable was set and it was different from where GIT_DIR appears as ".git" at its top, but the logic was faulty when .git is a "gitdir:" file that points at the real place, causing trouble in working trees that are managed by "git worktree". This has been corrected.
- Codepaths that read from an on-disk loose object were too loose in validating that they are reading a proper object file and sometimes read past the data they read from the disk, which has been corrected. H/t to Gustavo Grieco for reporting.
- The original command line syntax for "git merge", which was "git merge <msg> HEAD <parent>...", has been deprecated for quite some time, and "git gui" was the last in-tree user of the syntax. This is finally fixed, so that we can move forward with the deprecation.
- An author name that has a backslash-quoted double quote in the human readable part ("My "double quoted" name"), was not unquoted correctly while applying a patch from a piece of e-mail.
- Doc update to clarify what "log -3 --reverse" does.
- Almost everybody uses DEFAULT_ABBREV to refer to the default setting for the abbreviation, but "git blame" peeked into underlying variable bypassing the macro for no good reason.
- The "graph" API used in "git log --graph" miscounted the number of output columns consumed so far when drawing a padding line, which has been fixed; this did not affect

any existing code as nobody tried to write anything after the padding on such a line, though.

- The code that parses the format parameter of the for-each-ref command has seen a micro-optimization.

- When we started to use cURL to talk to an imap server, we forgot to explicitly add imap(s): and the library tried to make HTTP(s) requests instead.

- The ./configure script generated from configure.ac was taught how to detect support of SSL by libcurl better.

- The command-line completion script (in contrib/) learned to complete "git cmd ^mas<HT>" to complete the negative end of reference to "git cmd ^master". (merge 49416ad22a cp/completion-negative-refs later to maint).

- The existing "git fetch --depth=<n>" option was hard to use correctly when making the history of an existing shallow clone deeper. A new option, "--deepen=<n>", has been added to make this easier to use. "git clone" also learned "--shallow-since=<date>" and "--shallow-exclude=<tag>" options to make it easier to specify "I am interested only in the recent N months worth of history" and "Give me only the history since that version". (merge cccf74e2da nd/shallow-deepen later to maint).

- "git blame --reverse OLD path" is now DWIMmed to show how lines in path in an old revision OLD have survived up to the current commit. (merge e1d09701a4 jc/blame-reverse later to maint).

- The http.emptyauth configuration variable is a way to allow an empty username to pass when attempting to authenticate using mechanisms like Kerberos. We took an unspecified (NULL) username and sent ":" (i.e. no username, no password) to CURLOPT_USERPWD, but did not do the same when the username is explicitly set to an empty string.

- "git clone" of a local repository can be done at the filesystem level, but the codepath did not check errors while copying and adjusting the file that lists alternate object stores.

- Documentation for "git commit" was updated to clarify that "commit -p <paths>" adds to the current contents of the index to come up with what to commit.

- A stray symbolic link in the $GIT_DIR/refs/ directory could make name resolution loop forever, which has been corrected.

- The "submodule.<name>.path" stored in .gitmodules is never copied to .git/config and such a key in .git/config has no meaning, but the documentation described it next to submodule.<name>.url as if both belong to .git/config. This has been fixed.

- In a worktree created via "git worktree", "git checkout" attempts to protect users from confusion by refusing to check out a branch that is already checked out in another

worktree. However, this also prevented checking out a branch which is designated as the primary branch of a bare repository, in a worktree that is connected to the bare repository. The check has been corrected to allow it.

- "git rebase" immediately after "git clone" failed to find the fork point from the upstream.

- When fetching from a remote that has many tags that are irrelevant to branches we are following, we used to waste way too many cycles checking if the object pointed at by a tag (that we are not going to fetch!) exists in our repository too carefully.

- Protect our code from over-eager compilers.

- Recent git allows submodule.<name>.branch to use a special token "." instead of the branch name; the documentation has been updated to describe it.

- "git send-email" attempts to pick up valid e-mails from the trailers, but people in the real world write non-addresses there, like "Cc: Stable <add@re.ss> # 4.8+", which broke the output depending on the availability and vintage of the Mail::Address perl module. (merge dcfafc5214 mm/send-email-cc-cruft-after-address later to maint).

- The Travis CI configuration we ship ran the tests with the --verbose option but this risks non-TAP output that happens to be "ok" to be misinterpreted as TAP signalling a test that passed. This resulted in unnecessary failures. This has been corrected by introducing a new mode to run our tests in the test harness to send the verbose output separately to the log file.

- Some AsciiDoc formatters mishandle a displayed illustration with tabs in it. Adjust a few of them in merge-base documentation to work around them.

- Fixed a minor regression in "git submodule" that was introduced when more helper functions were reimplemented in C. (merge 77b63ac31e sb/submodule-ignore-trailing-slash later to maint).

- The code that we have used for the past 10+ years to cycle 4-element ring buffers turns out to be not quite portable in theoretical world. (merge bb84735c80 rs/ring-buffer-wraparound later to maint).

- "git daemon" used fixed-length buffers to turn URLs to the repository the client asked for into the server side directory paths, using snprintf() to avoid overflowing these buffers, but allowed possibly truncated paths to the directory. This has been tightened to reject such a request that causes an overlong path to be served. (merge 6bdb0083be jk/daemon-path-ok-check-truncation later to maint).

- Recent update to git-sh-setup (a library of shell functions that are used by our in-tree scripted Porcelain commands) included another shell library git-sh-i18n without specifying where it is, relying on the $PATH. This has been fixed to be more explicit by prefixing with $(git --exec-path) output. (merge 1073094f30 ak/sh-setup-dot-source-i18n-fix later to

maint).

- Fix for a racy false-positive test failure. (merge fdf4f6c79b as/merge-attr-sleep later to maint).

- Portability update and workaround for builds on recent Mac OS X. (merge a296bc0132 ls/macos-update later to maint).

- Using a %(HEAD) placeholder in "for-each-ref --format=" option caused the command to segfault when on an unborn branch. (merge 84679d470d jc/for-each-ref-head-segfault-fix later to maint).

- "git rebase -i" did not work well with the core.commentchar configuration variable for two reasons, both of which have been fixed. (merge 882cd23777 js/rebase-i-commentchar-fix later to maint).

- Other minor doc, test and build updates and code cleanups. (merge 5c238e29a8 jk/common-main later to maint). (merge 5a5749e45b ak/pre-receive-hook-template-modefix later to maint). (merge 6d834ac8f1 jk/rebase-config-insn-fmt-docfix later to maint). (merge de9f7fa3b0 rs/commit-pptr-simplify later to maint). (merge 4259d693fc sc/fmt-merge-msg-doc-markup-fix later to maint). (merge 28fab7b23d nd/test-helpers later to maint). (merge c2bb0c1d1e rs/cocci later to maint). (merge 3285b7badb ps/common-info-doc later to maint). (merge 2b090822e8 nd/worktree-lock later to maint). (merge 4bd488ea7c jk/create-branch-remove-unused-param later to maint). (merge 974e0044d6 tk/diffcore-delta-remove-unused later to maint).

□□□ #2
□ □□□ □□□ 2026-10-31 19:53:46 UTC
□ □□□ □□□ 2026-10-31 19:53:49 UTC