1.8.5

Git v1.8.5 Release Notes ========================

Backward compatibility notes (for Git 2.0) ----------

When "git push [$there]" does not say what to push, we have used the traditional "matching" semantics so far (all your branches were sent to the remote as long as there already are branches of the same name over there). In Git 2.0, the default will change to the "simple" semantics, which pushes:

- only the current branch to the branch with the same name, and only when the current branch is set to integrate with that remote branch, if you are pushing to the same remote as you fetch from; or

- only the current branch to the branch with the same name, if you are pushing to a remote that is not where you usually fetch from.

Use the user preference configuration variable "push.default" to change this. If you are an old-timer who is used to the "matching" semantics, you can set the variable to "matching" to keep the traditional behaviour. If you want to live in the future early, you can set it to "simple" today without waiting for Git 2.0.

When "git add -u" (and "git add -A") is run inside a subdirectory and does not specify which paths to add on the command line, it will operate on the entire tree in Git 2.0 for consistency with "git commit -a" and other commands. There will be no mechanism to make plain "git add -u" behave like "git add -u .". Current users of "git add -u" (without a pathspec) should start training their fingers to explicitly say "git add -u ." before Git 2.0 comes. A warning is issued when these commands are run without a pathspec and when you have local changes outside the current directory, because the behaviour in Git 2.0 will be different from today’s version in such a situation.

In Git 2.0, "git add <path>" will behave as "git add -A <path>", so that "git add dir/" will notice paths you removed from the directory and record the removal. Versions before Git 2.0, including this release, will keep ignoring removals, but the users who rely on this behaviour are encouraged to start using "git add --ignore-removal <path>" now before 2.0 is released.

The default prefix for "git svn" will change in Git 2.0. For a long time, "git svn" created its remote-tracking branches directly under refs/remotes, but it will place them under refs/remotes/origin/ unless it is told otherwise with its --prefix option.

Updates since v1.8.4 ----------

Foreign interfaces, subsystems and ports.

- "git-svn" has been taught to use the serf library, which is the only option SVN 1.8.0 offers us when talking the HTTP protocol.

- "git-svn" talking over an https:// connection using the serf library dumped core due to a bug in the serf library that SVN uses. Work around it on our side, even though the SVN side is being fixed.

- On MacOS X, we detected if the filesystem needs the "pre-composed unicode strings" workaround, but did not automatically enable it. Now we do.

- remote-hg remote helper misbehaved when interacting with a local Hg repository relative to the home directory, e.g. "clone hg::~/there".

- imap-send ported to OS X uses Apple’s security framework instead of OpenSSL’s.

- "git fast-import" treats an empty path given to "ls" as the root of the tree.

UI, Workflows & Features

- xdg-open can be used as a browser backend for "git web-browse" (hence to show "git help -w" output), when available.

- "git grep" and "git show" pay attention to the "--textconv" option when these commands are told to operate on blob objects (e.g. "git grep -e pattern --textconv HEAD:Makefile").

- "git replace" helper no longer allows an object to be replaced with another object of a different type to avoid confusion (you can still manually craft such a replacement using "git update-ref", as an escape hatch).

- "git status" no longer prints the dirty status information of submodules for which submodule.$name.ignore is set to "all".

- "git rebase -i" honours core.abbrev when preparing the insn sheet for editing.

- "git status" during a cherry-pick shows which original commit is being picked.

- Instead of typing four capital letters "HEAD", you can say "@" now, e.g. "git log @".

- "git check-ignore" follows the same rule as "git add" and "git status" in that the ignore/exclude mechanism does not take effect on paths that are already tracked. With the "--no-index" option, it can be used to diagnose which paths that should have been ignored have been mistakenly added to the index.

- Some irrelevant "advice" messages that are shared with "git status" output have been removed from the commit log template.

- "update-refs" learned a "--stdin" option to read multiple update requests and perform them in an all-or-none fashion.

- Just like "make -C <directory>", "git -C <directory> ..." tells Git to go there before doing anything else.

- Just like "git checkout -" knows to check out, and "git merge -" knows to merge, the branch you were previously on, "git cherry-pick" now understands "git cherry-pick -" to pick from the previous branch.

- "git status" now omits the prefix to make its output a comment in a commit log editor, which is not necessary for human consumption. Scripts that parse the output of "git status" are advised to use "git status --porcelain" instead, as its format is stable and easier to parse.

- The ref syntax "foo^{tag}" (with the literal string "{tag}") peels a tag ref to itself, i.e. it's a no-op., and fails if "foo" is not a tag. "git rev-parse --verify v1.0^{tag}" is a more convenient way than "test $(git cat-file -t v1.0) = tag" to check if v1.0 is a tag.

- "git branch -v -v" (and "git status") did not distinguish among a branch that is not based on any other branch, a branch that is in sync with its upstream branch, and a branch that is configured with an upstream branch that no longer exists.

- Earlier we started rejecting any attempt to add the 0{40} object name to the index and to tree objects, but it sometimes is necessary to allow this to be able to use tools like filter-branch to correct such broken tree objects. "filter-branch" can again be used to do this.

- "git config" did not provide a way to set or access numbers larger than a native "int" on the platform; it now provides 64-bit signed integers on all platforms.

- "git pull --rebase" always chose to do the bog-standard flattening rebase. You can tell it to run "rebase --preserve-merges" with "git pull --rebase=preserve" or by setting "pull.rebase" configuration to "preserve".

- "git push --no-thin" actually disables the "thin pack transfer" optimization.

- Magic pathspecs like ":(icase)makefile" (matches both Makefile and makefile) and ":(glob)foo/**/bar" (matches "bar" in "foo" and any subdirectory of "foo") can be used in more places.

- The "http.*" variables can now be specified for individual URLs. For example,

```
[http]
    sslVerify = true
[http "https://weak.example.com/"]
    sslVerify = false
```

```
would flip http.sslVerify off only when talking to that specific
site.
```

- "git mv A B" when moving a submodule has been taught to relocate the submodule’s working tree and to adjust the paths in the .gitmodules file.
- "git blame" can now take more than one -L option to discover the origin of multiple blocks of lines.
- The http transport clients can optionally ask to save cookies with the http.savecookies configuration variable.
- "git push" learned a more fine grained control over a blunt "--force" when requesting a non-fast-forward update with the "--force-with-lease=<refname>:<expected object name>" option.
- "git diff --diff-filter=<classes of changes>" can now take lowercase letters (e.g. "--diff-filter=d") to mean "show everything but these classes". "git diff-files -q" is now a deprecated synonym for "git diff-files --diff-filter=d".
- "git fetch" (hence "git pull" as well) learned to check "fetch.prune" and "remote.*.prune" configuration variables and to behave as if the "--prune" command line option was given.
- "git check-ignore -z" applied the NUL termination to both its input (with --stdin) and its output, but "git check-attr -z" ignored the option on the output side. Make both honor -z on the input and output side the same way.
- "git whatchanged" may still be used by old timers, but mention of it in documents meant for new users will only waste readers’ time wondering what the difference is between it and "git log". Make it less prominent in the general part of the documentation and explain that it is merely a "git log" with different default behaviour in its own document.

Performance, Internal Implementation, etc.

- "git for-each-ref" when asking for merely the object name does not have to parse the object pointed at by the refs; the codepath has been optimized.
- The HTTP transport will try to use TCP keepalive when able.
- "git repack" is now written in C.
- Build procedure for MSVC has been updated.
- If a build-time fallback is set to "cat" instead of "less", we should apply the same "no subprocess or pipe" optimization as we apply to user-supplied GIT_PAGER=cat.

- Many commands use a --dashed-option as an operation mode selector (e.g. "git tag --delete") that excludes other operation modes (e.g. "git tag --delete --verify" is nonsense) and that cannot be negated (e.g. "git tag --no-delete" is nonsense). The parse-options API learned a new OPT_CMDMODE macro to make it easier to implement such a set of options.

- OPT_BOOLEAN() in the parse-options API was misdesigned to be "counting up" but many subcommands expect it to behave as "on/off". Update them to use OPT_BOOL() which is a proper boolean.

- "git gc" exits early without doing any work when it detects that another instance of itself is already running.

- Under memory pressure and/or file descriptor pressure, we used to close pack windows that are not used and also closed filehandles to open but unused packfiles. These are now controlled separately to better cope with the load.

Also contains various documentation updates and code clean-ups.

Fixes since v1.8.4 ----------

Unless otherwise noted, all the fixes since v1.8.4 in the maintenance track are contained in this release (see the maintenance releases' notes for details).

- An ancient How-To on serving Git repositories on an HTTP server lacked a warning that it has been mostly superseded with a more modern way. (merge 6d52bc3 sc/doc-howto-dumb-http later to maint).

- The interaction between the use of Perl in our test suite and NO_PERL has been clarified a bit. (merge f8fc0ee jn/test-prereq-perl-doc later to maint).

- The synopsis section of the "git unpack-objects" documentation has been clarified a bit. (merge 61e2e22 vd/doc-unpack-objects later to maint).

- We did not generate the HTML version of the documentation to "git subtree" in contrib/. (merge 95c62fb jk/subtree-install-fix later to maint).

- A fast-import stream expresses a pathname with funny characters by quoting them in C style; the remote-hg remote helper forgot to unquote such a path. (merge 1136265 ap/remote-hg-unquote-cquote later to maint).

- "git reset -p HEAD" has a codepath to special-case it to behave differently from resetting to contents of other commits, but a recent change broke it.

- Coloring around octopus merges in "log --graph" output was screwy. (merge 339c17b hn/log-graph-color-octopus later to maint).

- "git checkout topic", when there is not yet a local "topic" branch but there is a unique remote-tracking branch for a remote "topic" branch, pretended as if "git checkout -t -b topic remote/$r/topic" (for that unique remote $r) was run. This hack however was not implemented for "git checkout topic --". (merge bca3969 mm/checkout-auto-track-fix later to maint).

- One long-standing flaw in the pack transfer protocol used by "git clone" was that there was no way to tell the other end which branch "HEAD" points at, and the receiving end needed to guess. A new capability has been defined in the pack protocol to convey this information so that cloning from a repository with more than one branch pointing at the same commit where the HEAD is at now reliably sets the initial branch in the resulting repository. (merge 360a326 jc/upload-pack-send-symref later to maint).

- We did not handle cases where the http transport gets redirected during the authorization request (e.g. from http:// to https://). (merge 70900ed jk/http-auth-redirects later to maint).

- Bash prompting code to deal with an SVN remote as an upstream was coded in a way unsupported by older Bash versions (3.x). (merge 52ec889 sg/prompt-svn-remote-fix later to maint).

- The fall-back parsing of commit objects with broken author or committer lines was less robust than ideal in picking up the timestamps. (merge 03818a4 jk/split-broken-ident later to maint).

- "git rev-list --objects $^{v1.0}$ v1.0" gave the v1.0 tag itself in the output, but "git rev-list --objects v1.0^..v1.0" did not. (merge 895c5ba jc/revision-range-unpeel later to maint).

- "git clone" wrote some progress messages to standard output, not to standard error, and did not suppress them with the --no-progress option. (merge 643f918 jk/clone-progress-to-stderr later to maint).

- "format-patch --from=<whom>" forgot to omit an unnecessary in-body from line, i.e. when <whom> is the same as the real author. (merge 662cc30 jk/format-patch-from later to maint).

- "git shortlog" used to choke and die when there is a malformed commit (e.g. missing authors); it now simply ignores such a commit and keeps going. (merge cd4f09e jk/shortlog-tolerate-broken-commit later to maint).

- "git merge-recursive" did not parse its "--diff-algorithm=" command line option correctly. (merge 6562928 jk/diff-algo later to maint).

- When running "fetch -q", a long silence while the sender side computes the set of objects to send can be mistaken by proxies as dropped connection. The server side has been taught to send a small empty messages to keep the connection alive. (merge 115dedd

jk/upload-pack-keepalive later to maint).

- "git rebase" had a portability regression in v1.8.4 that triggered a bug in some BSD shell implementations. (merge 99855dd mm/rebase-continue-freebsd-WB later to maint).

- "git branch --track" had a minor regression in v1.8.3.2 and later that made it impossible to base your local work on anything but a local branch of the upstream repository you are tracking. (merge b0f49ff jh/checkout-auto-tracking later to maint).

- When the web server responds with "405 Method Not Allowed", "git http-backend" should tell the client what methods are allowed with the "Allow" header. (merge 9247be0 bc/http-backend-allow-405 later to maint).

- When there is no sufficient overlap between old and new history during a "git fetch" into a shallow repository, objects that the sending side knows the receiving end has were unnecessarily sent. (merge f21d2a7 nd/fetch-into-shallow later to maint).

- "git cvsserver" computed the permission mode bits incorrectly for executable files. (merge 1b48d56 jc/cvsserver-perm-bit-fix later to maint).

- When send-email obtains an error message to die with upon failure to start an SSL session, it tried to read the error string from a wrong place. (merge 6cb0c88 bc/send-email-ssl-die-message-fix later to maint).

- The implementation of "add -i" has some crippling code to work around an ActiveState Perl limitation but it by mistake also triggered on Git for Windows where MSYS perl is used. (merge df17e77 js/add-i-mingw later to maint).

- We made sure that we notice when the user-supplied GIT_DIR is actually a gitfile, but did not do the same when the default ".git" is a gitfile. (merge 487a2b7 nd/git-dir-pointing-at-gitfile later to maint).

- When an object is not found after checking the packfiles and the loose object directory, read_sha1_file() re-checks the packfiles to prevent racing with a concurrent repacker; teach the same logic to has_sha1_file(). (merge 45e8a74 jk/has-sha1-file-retry-packed later to maint).

- "git commit --author=$name", when $name is not in the canonical "A. U. Thor <au.thor@example.xz>" format, looks for a matching name from existing history, but did not consult mailmap to grab the preferred author name. (merge ea16794 ap/commit-author-mailmap later to maint).

- "git ls-files -k" needs to crawl only the part of the working tree that may overlap the paths in the index to find killed files, but shared code with the logic to find all the untracked files, which made it unnecessarily inefficient. (merge 680be04 jc/ls-files-killed-optim later to maint).

- The shortened commit object names in the insn sheet that is prepared at the beginning of a "rebase -i" session can become ambiguous as the rebasing progresses and the repository gains more commits. Make sure the internal record is kept with full 40-hex object names. (merge 75c6976 es/rebase-i-no-abbrev later to maint).

- "git rebase --preserve-merges" internally used the merge machinery and as a side effect left the merge summary message in the log, but when rebasing there is no need for the merge summary. (merge a9f739c rt/rebase-p-no-merge-summary later to maint).

- A call to xread() was used without a loop around it to cope with short reads in the codepath to stream new contents to a pack. (merge e92527c js/xread-in-full later to maint).

- "git rebase -i" forgot that the comment character is configurable while reading its insn sheet. (merge 7bca7af es/rebase-i-respect-core-commentchar later to maint).

- The mailmap support code read past the allocated buffer when the mailmap file ended with an incomplete line. (merge f972a16 jk/mailmap-incomplete-line later to maint).

- We used to send a large request to read(2)/write(2) as a single system call, which was bad from the latency point of view when the operation needs to be killed, and also triggered an error on broken 64-bit systems that refuse to read or write more than 2GB in one go. (merge a487916 sp/clip-read-write-to-8mb later to maint).

- "git fetch" that auto-followed tags incorrectly reused the connection with Git-aware transport helper (like the sample "ext::" helper shipped with Git). (merge 0f73f8b jc/transport-do-not-use-connect-twice-in-fetch later to maint).

- "git log --full-diff — <pathspec>" showed a huge diff for paths outside the given <pathspec> for each commit, instead of showing the change relative to the parent of the commit. "git reflog -p" had a similar problem. (merge 838f9a1 tr/log-full-diff-keep-true-parents later to maint).

- Setting a submodule.*.path configuration variable to true (without giving "= value") caused Git to segfault. (merge 4b05440 jl/some-submodule-config-are-not-boolean later to maint).

- "git rebase -i" (there could be others, as the root cause is pretty generic) fed a random, data dependent string to ’echo’ and expected it to come out literally, corrupting its error message. (merge 89b0230 mm/no-shell-escape-in-die-message later to maint).

- Some people still use rather old versions of bash, which cannot grok some constructs like ’printf -v varname’ which the prompt and completion code started to use recently. (merge a44aa69 bc/completion-for-bash-3.0 later to maint).

- Code to read configuration from a blob object did not compile on platforms with fgetc() etc. implemented as macros. (merge 49d6cfa hv/config-from-blob later to maint-1.8.3).

- The recent "short-cut clone connectivity check" topic broke a shallow repository when a fetch operation tries to auto-follow tags. (merge 6da8bdc nd/fetch-pack-shallow-fix later to maint-1.8.3).

□□□ #2
□ □□□ □□□ 2026-10-31 19:52:47 UTC
□ □□□ □□□ 2026-10-31 19:52:49 UTC