1.6.6

Git v1.6.6 Release Notes ========================

Notes on behaviour change ------------------------

- In this release, "git fsck" defaults to "git fsck --full" and checks packfiles, and because of this it will take much longer to complete than before. If you prefer a quicker check only on loose objects (the old default), you can say "git fsck --no-full". This has been supported by 1.5.4 and newer versions of git, so it is safe to write it in your script even if you use slightly older git on some of your machines.

Preparing yourselves for compatibility issues in 1.7.0 --

In git 1.7.0, which is planned to be the release after 1.6.6, there will be a handful of behaviour changes that will break backward compatibility.

These changes were discussed long time ago and existing behaviours have been identified as more problematic to the userbase than keeping them for the sake of backward compatibility.

When necessary, a transition strategy for existing users has been designed not to force them running around setting configuration variables and updating their scripts in order to either keep the traditional behaviour or adjust to the new behaviour, on the day their sysadmin decides to install the new version of git. When we switched from "git-foo" to "git foo" in 1.6.0, even though the change had been advertised and the transition guide had been provided for a very long time, the users procrastinated during the entire transition period, and ended up panicking on the day their sysadmins updated their git installation. We are trying to avoid repeating that unpleasantness in the 1.7.0 release.

For changes decided to be in 1.7.0, commands that will be affected have been much louder to strongly discourage such procrastination, and they continue to be in this release. If you have been using recent versions of git, you would have seen warnings issued when you used features whose behaviour will change, with a clear instruction on how to keep the existing behaviour if you want to. You hopefully are already well prepared.

Of course, we have also been giving "this and that will change in 1.7.0; prepare yourselves" warnings in the release notes and announcement messages for the past few releases. Let’s see how well users will fare this time.

- "git push" into a branch that is currently checked out (i.e. pointed by HEAD in a repository that is not bare) will be refused by default.

```
Similarly, "git push $there :$killed" to delete the branch $killed
in a remote repository $there, when $killed branch is the current
branch pointed at by its HEAD, will be refused by default.

Setting the configuration variables receive.denyCurrentBranch and
receive.denyDeleteCurrent to 'ignore' in the receiving repository
can be used to override these safety features.  Versions of git
since 1.6.2 have issued a loud warning when you tried to do these
operations without setting the configuration, so repositories of
people who still need to be able to perform such a push should
already have been future proofed.

Please refer to:

https://archive.kernel.org/oldwiki/git.wiki.kernel.org/index.php/GitFaq.html#non-bare
https://lore.kernel.org/git/7vbptlsuyv.fsf@gitster.siamese.dyndns.org/

for more details on the reason why this change is needed and the
transition process that already took place so far.
```

- "git send-email" will not make deep threads by default when sending a patch series with more than two messages. All messages will be sent as a reply to the first message, i.e. cover letter. Git 1.6.6 (this release) will issue a warning about the upcoming default change, when it uses the traditional "deep threading" behaviour as the built-in default. To squelch the warning but still use the "deep threading" behaviour, give --chain-reply-to option or set sendemail.chainreplyto to true.

```
It has been possible to configure send-email to send "shallow thread"
by setting sendemail.chainreplyto configuration variable to false.
The only thing 1.7.0 release will do is to change the default when
you haven't configured that variable.
```

- "git status" will not be "git commit --dry-run". This change does not affect you if you run the command without pathspec.

```
Nobody sane found the current behaviour of "git status Makefile" useful
nor meaningful, and it confused users.  "git commit --dry-run" has been
provided as a way to get the current behaviour of this command since
1.6.5.
```

- "git diff" traditionally treated various "ignore whitespace" options only as a way to filter the patch output. "git diff --exit-code -b" exited with non-zero status even if all changes were about changing the amount of whitespace and nothing else. and "git diff -b" showed the "diff --git" header line for such a change without patch text.

```
In 1.7.0, the "ignore whitespaces" will affect the semantics of the
diff operation itself.  A change that does not affect anything but
whitespaces will be reported with zero exit status when run with
--exit-code, and there will not be "diff --git" header for such a
change.
```

Updates since v1.6.5 ----------

(subsystems)

- various gitk updates including use of themed widgets under Tk 8.5, Japanese translation, a fix to a bug when running "gui blame" from a subdirectory, etc.
- various git-gui updates including new translations, wm states fixes, Tk bug workaround after quitting, improved heuristics to trigger gc, etc.
- various git-svn updates.
- "git fetch" over http learned a new mode that is different from the traditional "dumb commit walker".

(portability)

- imap-send can be built on mingw port.

(performance)

- "git diff -B" has smaller memory footprint.

(usability, bells and whistles)

- The object replace mechanism can be bypassed with --no-replace-objects global option given to the "git" program.
- In configuration files, a few variables that name paths can begin with ~/ and ~username/ and they are expanded as expected.
- "git subcmd -h" now shows short usage help for many more subcommands.
- "git bisect reset" can reset to an arbitrary commit.

- "git checkout frotz" when there is no local branch "frotz" but there is only one remote tracking branch "frotz" is taken as a request to start the named branch at the corresponding remote tracking branch.

- "git commit -c/-C/--amend" can be told with a new "--reset-author" option to ignore authorship information in the commit it is taking the message from.

- "git describe" can be told to add "-dirty" suffix with "--dirty" option.

- "git diff" learned --submodule option to show a list of one-line logs instead of differences between the commit object names.

- "git diff" learned to honor diff.color.func configuration to paint function name hint printed on the hunk header "@@ -j,k +l,m @@" line in the specified color.

- "git fetch" learned --all and --multiple options, to run fetch from many repositories, and --prune option to remove remote tracking branches that went stale. These make "git remote update" and "git remote prune" less necessary (there is no plan to remove "remote update" nor "remote prune", though).

- "git fsck" by default checks the packfiles (i.e. "--full" is the default); you can turn it off with "git fsck --no-full".

- "git grep" can use -F (fixed strings) and -i (ignore case) together.

- import-tars contributed fast-import frontend learned more types of compressed tarballs.

- "git instaweb" knows how to talk with mod_cgid to apache2.

- "git log --decorate" shows the location of HEAD as well.

- "git log" and "git rev-list" learned to take revs and pathspecs from the standard input with the new "--stdin" option.

- "--pretty=format" option to "log" family of commands learned:

1. to wrap text with the "%w()" specifier.
2. to show reflog information with "%g[sdD]" specifier.

- "git notes" command to annotate existing commits.

- "git merge" (and "git pull") learned --ff-only option to make it fail if the merge does not result in a fast-forward.

- "git mergetool" learned to use p4merge.

- "git rebase -i" learned "reword" that acts like "edit" but immediately starts an editor to tweak the log message without returning control to the shell, which is done by "edit" to give an opportunity to tweak the contents.

- "git send-email" can be told with "--envelope-sender=auto" to use the same address as "From:" address as the envelope sender address.

- "git send-email" will issue a warning when it defaults to the --chain-reply-to behaviour without being told by the user and instructs to prepare for the change of the default in 1.7.0 release.

- In "git submodule add <repository> <path>", <path> is now optional and inferred from <repository> the same way "git clone <repository>" does.

- "git svn" learned to read SVN 1.5+ and SVK merge tickets.

- "git svn" learned to recreate empty directories tracked only by SVN.

- "gitweb" can optionally render its "blame" output incrementally (this requires JavaScript on the client side).

- Author names shown in gitweb output are links to search commits by the author.

Fixes since v1.6.5 ----------

All of the fixes in v1.6.5.X maintenance series are included in this release, unless otherwise noted.

□□□ #2
□ □□□ □□□ 2026-10-31 19:48:28 UTC
□ □□□ □□□ 2026-10-31 19:48:29 UTC