git-lfs /
git-lfs

Code | Issues 383 | Pull requests 19 | Discussions | Actions | Projects 1 | Wiki | Security 3 | Insight

Git extension for versioning large files

git-lfs.com

View license

Code of conduct

Security policy

13.1k stars | 2k forks | 485 watching | Branches | Activity | Custom properties | Tags

Public repository

... | 86 Branches | 105 Tags | Go to file t | Go to file | Add file | Code

chrisd8088 Merge pull request #5930 from bk2204/tighten-actions-security ✓ a32a02b · 5 days ago

.github	workflows: don't persist credentials	5 days ago
commands	Merge branch 'main' into main	2 months ago
config	release: v3.6.0	last month
creds	Implement multistage authentication	6 months ago
debian	release: v3.6.0	last month
docker	docker/README.md: update and simplify text	last year
docs	docs/howto/release-git-lfs.md: add Go upgr...	3 weeks ago
errors	errors/types.go: correctly wrap retriable errors	6 months ago
filepathfilter	make additional message strings translatable	2 years ago
fs	Replace deprecated `io/ioutil` functions	11 months ago
git	git: improve sparse file support	last month
lfs	hook: fix newlines in command missing mess...	2 months ago
lfsapi	lfsapi/lfsapi.go: enhance trace logs	last month
lfshttp	lfshttp,tq,t: don't fail on retriable batch errors	6 months ago
locking	ssh: add support for lazy spawning of pure S...	11 months ago
po	Introduce a single translation	3 years ago
rpm	release: v3.6.0	last month
script	script/upload: use generic sha3sum input for...	3 weeks ago
ssh	ssh/connection.go: report error when no sess...	last month
subprocess	subprocess: add a version of BufferedExec wi...	7 months ago

t	Merge branch 'main' into test-grep-clone-fix...	last month
tasklog	Replace deprecated `io/ioutil` functions	11 months ago
tools	tools/iotools.go: always close temp spool file	11 months ago
tq	lfshttp,tq,t: don't fail on retriable batch errors	6 months ago
tr	Replace deprecated `io/ioutil` functions	11 months ago
.gitattributes	Enable autocrlf	9 years ago
.gitignore	gitignore: ignore the vendor directory	2 years ago
.mailmap	Add myself to .mailmap	7 years ago
.ruby-version	.github/workflows: upgrade to ruby/setup-ru...	2 years ago
CHANGELOG.md	release: v3.6.0	last month
CODE-OF-CONDUCT.md	embed the open code of conduct since the li...	8 years ago
CONTRIBUTING.md	Remove vendoring instructions from contrib...	last year
INSTALLING.md	Corrected Pop!_OS naming	last week
LICENSE.md	LICENSE.md: update license years and vendo...	3 weeks ago
Makefile	Makefile: split release signing workflow for ...	11 months ago
README.md	README.md: note current minimum tested G...	3 weeks ago
SECURITY.md	{README,SECURITY}.md: add security bug re...	4 years ago
git-lfs.go	make additional message strings translatable	2 years ago
git-lfs_windows.go	Add support for Windows arm64	3 years ago
git-lfs_windows_arm64.go	Fix Windows arm64 build	3 years ago
go.mod	go.mod: require 1.21	6 months ago
go.sum	build(deps): bump golang.org/x/net from 0.1...	8 months ago
versioninfo.json	release: v3.6.0	last month

Git Large File Storage

Git LFS is a command line extension and specification for managing large files with Git.

The client is written in Go, with pre-compiled binaries available for Mac, Windows, Linux, and FreeBSD. Check out the website for an overview of features.

Getting Started

Installing

On Linux

Debian and RPM packages are available from packagecloud, see the Linux installation instructions.

On macOS

Homebrew bottles are distributed and can be installed via `brew install git-lfs`.

On Windows

Git LFS is included in the distribution of Git for Windows. Alternatively, you can install a recent version of Git LFS from the Chocolatey package manager.

From binary

Binary packages are available for Linux, macOS, Windows, and FreeBSD. The binary packages include a script which will:

- Install Git LFS binaries onto the system `$PATH`. On Windows in particular, you may need to restart your command shell so any change to `$PATH` will take effect and Git can locate the Git LFS binary.
- Run `git lfs install` to perform required global configuration changes.

```
$ ./install.sh
```

Note that Debian and RPM packages are built for multiple Linux distributions and versions for both amd64 and i386. For arm64, only Debian packages are built and only for recent versions due to the cost of building in emulation.

From source

- Ensure you have the latest version of Go, GNU make, and a standard Unix-compatible build environment installed.
- On Windows, install `goversioninfo` with `go install github.com/josephspurrier/goversioninfo/cmd/goversioninfo@latest`.
- Run `make`.
- Place the `git-lfs` binary, which can be found in `bin`, on your system's executable `$PATH` or equivalent.
- Git LFS requires global configuration changes once per-machine. This can be done by running: `git lfs install`

Verifying releases

Releases are signed with the OpenPGP key of one of the core team members. To get these keys, you can run the following command, which will print them to standard output:

```
$ curl -L https://api.github.com/repos/git-lfs/git-lfs/tarball/core-gpg-keys | tar -Ozxf -
```

Once you have the keys, you can download the `sha256sums.asc` file and verify the file you want like so:

```
$ gpg -d sha256sums.asc | grep git-lfs-linux-amd64-v2.10.0.tar.gz | shasum -a 256 -c
```

For the convenience of distributors, we also provide a wider variety of signed hashes in the `hashes.asc` file. Those hashes are in the tagged BSD format, but can be verified with Perl's `shasum` or the GNU hash utilities, just like the ones in `sha256sums.asc`.

Example Usage

To begin using Git LFS within a Git repository that is not already configured for Git LFS, you can indicate which files you would like Git LFS to manage. This can be done by running the following *from within a Git repository*:

```
$ git lfs track "*.psd"
```

(Where `*.psd` is the pattern of filenames that you wish to track. You can read more about this pattern syntax here).

> *Note:* the quotation marks surrounding the pattern are important to prevent the glob pattern from being expanded by the shell.

After any invocation of `git-lfs-track(1)` or `git-lfs-untrack(1)`, you *must commit changes to your `.gitattributes` file*. This can be done by running:

```
$ git add .gitattributes
$ git commit -m "track *.psd files using Git LFS"
```

You can now interact with your Git repository as usual, and Git LFS will take care of managing your large files. For example, changing a file named `my.psd` (tracked above via `*.psd`):

```
$ git add my.psd
$ git commit -m "add psd"
```

> *Tip:* if you have large files already in your repository's history, `git lfs track` will *not* track them retroactively. To migrate existing large files in your history to use Git LFS, use `git lfs migrate`. For example:
>
> ```
> $ git lfs migrate import --include="*.psd" --everything
> ```
>
> **Note that this will rewrite history and change all of the Git object IDs in your repository, just like the export version of this command.**
>
> For more information, read `git-lfs-migrate(1)`.

You can confirm that Git LFS is managing your PSD file:

```
$ git lfs ls-files
3c2f7aedfb * my.psd
```

Once you've made your commits, push your files to the Git remote:

```
$ git push origin main
Uploading LFS objects: 100% (1/1), 810 B, 1.2 KB/s
# ...
To https://github.com/git-lfs/git-lfs-test
   67fcf6a..47b2002  main -> main
```

Uninstalling

If you've decided that Git LFS isn't right for you, you can convert your repository back to a plain Git repository with `git lfs migrate` as well. For example:

```
$ git lfs migrate export --include="*.psd" --everything
```

Note that this will rewrite history and change all of the Git object IDs in your repository, just like the import version of this command.

If there's some reason that things aren't working out for you, please let us know in an issue, and we'll definitely try to help or get it fixed.

Limitations

Git LFS maintains a list of currently known limitations, which you can find and edit here.

Current releases of Git LFS will work with Git versions as early as Git 2.0.0. However, for best performance, using a recent version of Git is highly recommended.

Git LFS source code utilizes Go modules in its build system, and therefore this project contains a `go.mod` file with a defined Go module path. However, we do not maintain a stable Go language API or ABI, as Git LFS is intended to be used solely as a compiled binary utility. Please do not import the `git-lfs` module into other Go code and do not rely on it as a source code dependency.

Need Help?

You can get help on specific commands directly:

```
$ git lfs help <subcommand>
```

The official documentation has command references and specifications for the tool. There's also a FAQ shipped with Git LFS which answers some common questions.

If you have a question on how to use Git LFS, aren't sure about something, or are looking for input from others on tips about best practices or use cases, feel free to start a discussion.

You can always open an issue, and one of the Core Team members will respond to you. Please be sure to include:

1. The output of `git lfs env`, which displays helpful information about your Git repository useful in debugging.
2. Any failed commands re-run with `GIT_TRACE=1` in the environment, which displays additional information pertaining to why a command crashed.

Contributing

See CONTRIBUTING.md for info on working on Git LFS and sending patches. Related projects are listed on the Implementations wiki page.

See also SECURITY.md for info on how to submit reports of security vulnerabilities.

Core Team

These are the humans that form the Git LFS core team, which runs the project.

In alphabetical order:

@chrisd8088	@larsxschneider
	
PGP 088335A9	PGP A5795889

Alumni

These are the humans that have in the past formed the Git LFS core team, or have otherwise contributed a significant amount to the project. Git LFS would not be possible without them.

In alphabetical order:

@andyneff	@bk2204	@PastelMobileSuit	@rubyist	@sinbad	@technoweenie	@ttaylorr
						

Releases 92

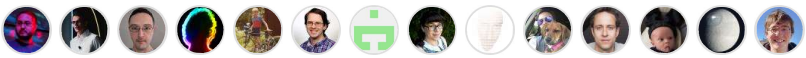
v3.6.0 Latest
3 weeks ago

+ 91 releases

Packages

No packages published

Contributors 209



+ 195 contributors

Languages

Go 61.8% Shell 35.9% Makefile 1.3% Other 1.0%