



Git Large File Storage

[Docs](#) [Downloads](#) [Source](#)

An open source Git extension for versioning large files

Git Large File Storage (LFS) replaces large files such as audio samples, videos, datasets, and graphics with text pointers inside Git, while storing the file contents on a remote server like GitHub.com or GitHub Enterprise.

 **Download** v2.12.1 (Windows)

How it works:



Git LFS security update:

[Windows users should update to 2.12.1 or newer.](#)

Getting Started

1.

[Download](#) and install the Git command line extension. Once downloaded and installed, set up Git LFS for your user account by running:

```
git lfs install
```

You only need to run this once per user account.

2.

In each Git repository where you want to use Git LFS, select the file types you'd like Git LFS to manage (or directly edit your `.gitattributes`). You can configure additional file extensions at anytime.

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

Now make sure `.gitattributes` is tracked:

```
git add .gitattributes
```

Note that defining the file types Git LFS should track will not, by itself, convert any pre-existing files to Git LFS, such as files on other branches or in your prior commit history. To do that, use the [git lfs migrate\[1\]](#) command, which has a range of options designed to suit various potential use cases.

3.

There is no step three. Just commit and push to GitHub as you normally would.

```
git add file.psd
git commit -m "Add design file"
git push origin master
```

Git LFS is an open source project

To file an issue or contribute to the project, head over [to the repository](#) or read our [guide to contributing](#).

If you're interested in integrating Git LFS into another tool or product, you might want to read the [API specification](#) or check out our [reference server implementation](#).

Features



Large file versioning

Version large files—even those as large as a couple GB in size—with Git.



More repository space

Host more in your Git repositories. External file storage makes it easy to keep your repository at a manageable size.



Faster cloning and fetching

Download less data. This means faster cloning and fetching from repositories that deal with large files.



Same Git workflow

Work like you always do on Git—no need for additional commands, secondary storage systems, or toolsets.



Same access controls and permissions

Keep the same access controls and permissions for large files as the rest of your Git repository when working with a remote host like GitHub.

⌘ with ❤ by **GitHub**.

Git is a trademark of the [Software Freedom Conservancy](#).