

NAME

git-lfs-checkout – Update working copy with file content if available

SYNOPSIS

git lfs checkout *filespec*... **git lfs checkout** **—to** *path* { **—ours** | **—theirs** | **—base** } *file*...

DESCRIPTION

Try to ensure that the working copy contains file content for Git LFS objects for the current ref, if the object data is available. Does not download any content, see **git-lfs-fetch(1)** for that.

Checkout scans the current ref for all LFS objects that would be required, then where a file is either missing in the working copy, or contains placeholder pointer content with the same SHA, the real file content is written, provided we have it in the local store. Modified files are never overwritten.

Filespecs can be provided as arguments to restrict the files which are updated.

When used with **—to** and the working tree is in a conflicted state due to a merge, this option checks out one of the three stages of the conflict into a separate file. This can make using diff tools to inspect and resolve merges easier.

OPTIONS

—base Check out the merge base of the specified file.

—ours Check out our side (that of the current branch) of the conflict for the specified file.

—theirs

Check out their side (that of the other branch) of the conflict for the specified file.

—to *path*

If the working tree is in a conflicted state, check out the portion of the conflict specified by **—base**, **—ours**, or **—theirs** to the given path.

EXAMPLES

- Checkout all files that are missing or placeholders

git lfs checkout

- Checkout a specific couple of files

git lfs checkout path/to/file1.png path/to.file2.png

SEE ALSO

git-lfs-fetch(1), **git-lfs-pull(1)**.

Part of the **git-lfs(1)** suite.