git-hash-object

git-hash-object(1) ==========

NAME ---- git-hash-object - Compute object ID and optionally create an object from a file

SYNOPSIS --------

> 'git hash-object' [-t <type>] [-w] [--path=<file> | --no-filters]
> [--stdin [--literally]] [--] <file>...
> 'git hash-object' [-t <type>] [-w] --stdin-paths [--no-filters]

DESCRIPTION ----------- Computes the object ID value for an object with specified type with the contents of the named file (which can be outside of the work tree), and optionally writes the resulting object into the object database. Reports its object ID to its standard output. When <type> is not specified, it defaults to "blob".

OPTIONS -------

-t <type>
Specify the type of object to be created (default: "blob"). Possible values are `commit`, `tree`, `blob`, and `tag`.

-w
Actually write the object into the object database.

--stdin
Read the object from standard input instead of from a file.

--stdin-paths
Read file names from the standard input, one per line, instead of from the command-line.

--path
Hash object as if it were located at the given path. The location of the file does not directly influence the hash value, but the path is used to determine which Git filters should be applied to the object before it can be placed in the object database. As a result of applying filters, the actual blob put into the object database may differ from the given file. This option is mainly useful for hashing temporary files located outside of the working directory or files read from stdin.

--no-filters
Hash the contents as is, ignoring any input filter that would have been chosen by the attributes

mechanism, including the end-of-line conversion. If the file is read from standard input then this is always implied, unless the `--path` option is given.

--literally
Allow `--stdin` to hash any garbage into a loose object which might not otherwise pass standard object parsing or git-fsck checks. Useful for stress-testing Git itself or reproducing characteristics of corrupt or bogus objects encountered in the wild.

GIT --- Part of the linkgit:git[1] suite

□□□ #1
□ □□□ □□□ 2026-10-31 20:11:11 UTC
□ □□□ □□□ 2026-10-31 20:11:11 UTC