



Rocky Enterprise Linux 9.2 Manual Pages on command 'git-commit-graph.1'

C:\>man git-commit-graph.1

GIT-COMMIT-GRAPH(1) Git Manual GIT-COMMIT-GRAPH(1)

NAME

git-commit-graph - Write and verify Git commit-graph files

SYNOPSIS

git commit-graph verify [--object-dir <dir>] [--shallow] [--[no-]progress]

git commit-graph write <options> [--object-dir <dir>] [--[no-]progress]

DESCRIPTION

Manage the serialized commit-graph file.

OPTIONS

--object-dir

Use given directory for the location of packfiles and commit-graph file. This parameter exists to specify the location of an alternate that only has the objects directory, not a full .git directory. The commit-graph file is expected to be in the <dir>/info directory and the packfiles are expected to be in <dir>/pack.

--[no-]progress

Turn progress on/off explicitly. If neither is specified, progress is shown if

standard error is connected to a terminal.

COMMANDS

write

Write a commit-graph file based on the commits found in packfiles.

With the `--stdin-packs` option, generate the new commit graph by walking objects only in the specified pack-indexes. (Cannot be combined with `--stdin-commits` or `--reachable`.)

With the `--stdin-commits` option, generate the new commit graph by walking commits starting at the commits specified in stdin as a list of OIDs in hex, one OID per line. (Cannot be combined with `--stdin-packs` or `--reachable`.)

With the `--reachable` option, generate the new commit graph by walking commits starting at all refs. (Cannot be combined with `--stdin-commits` or `--stdin-packs`.)

With the `--append` option, include all commits that are present in the existing commit-graph file.

With the `--split` option, write the commit-graph as a chain of multiple commit-graph files stored in `<dir>/info/commit-graphs`. The new commits not already in the commit-graph are added in a new "tip" file. This file is merged with the existing file if the following merge conditions are met:

? If `--size-multiple=<X>` is not specified, let X equal 2. If the new tip file would have N commits and the previous tip has M commits and X times N is greater than M, instead merge the two files into a single file.

? If `--max-commits=<M>` is specified with M a positive integer, and the new tip file would have more than M commits, then instead merge the new tip with the previous tip.

Finally, if `--expire-time=<datetime>` is not specified, let `datetime` be the current time. After writing the split commit-graph, delete all unused commit-graph whose modified times are older than `datetime`.

verify

Read the commit-graph file and verify its contents against the object database.

Used to check for corrupted data.

With the `--shallow` option, only check the tip commit-graph file in a chain of split commit-graphs.

EXAMPLES

? Write a commit-graph file for the packed commits in your local `.git` directory.

```
$ git commit-graph write
```

? Write a commit-graph file, extending the current commit-graph file using commits in `<pack-index>`.

```
$ echo <pack-index> | git commit-graph write --stdin-packs
```

? Write a commit-graph file containing all reachable commits.

```
$ git show-ref -s | git commit-graph write --stdin-commits
```

? Write a commit-graph file containing all commits in the current commit-graph file along with those reachable from HEAD.

```
$ git rev-parse HEAD | git commit-graph write --stdin-commits --append
```

GIT

