NAME

git-check-ignore - Debug gitignore / exclude files

SYNOPSIS

git check-ignore [options] pathname...
git check-ignore [options] --stdin < <list-of-paths>

DESCRIPTION

For each pathname given via the command-line or from a file via --stdin, show the pattern from .gitignore (or other input files to the exclude mechanism) that decides if the pathname is excluded or included. Later patterns within a file take precedence over earlier ones.

OPTIONS

-q, --quiet
: Don't output anything, just set exit status. This is only valid with a single pathname.

-v, --verbose
: Also output details about the matching pattern (if any) for each given pathname.

--stdin
: Read file names from stdin instead of from the command-line.

-z
: The output format is modified to be machine-parseable (see below). If --stdin is also given, input paths are separated with a NUL character instead of a linefeed character.

-n, --non-matching
: Show given paths which don't match any pattern. This only makes sense when --verbose is enabled, otherwise it would not be possible to distinguish between paths which match a pattern and those which don't.

--no-index
: Don't look in the index when undertaking the checks. This can be used to debug why a path became tracked by e.g. git add . and was not ignored by the rules as expected by the user or when developing patterns including negation to match a path previously added with git add -f.

OUTPUT

By default, any of the given pathnames which match an ignore pattern will be output, one per line. If no pattern matches a given path, nothing will be output for that path; this means that path will not be ignored.

If --verbose is specified, the output is a series of lines of the form:

<source> <COLON> <linenum> <COLON> <pattern> <HT> <pathname>

<pathname> is the path of a file being queried, <pattern> is the matching pattern, <source> is the pattern's source file, and <linenum> is the line number of the pattern within that source. If the pattern contained a ! prefix or / suffix, it will be preserved in the output. <source> will be an absolute path when referring to the file configured by core.excludesfile, or relative to the repository root when referring to .git/info/exclude or a per-directory exclude file.

If -z is specified, the pathnames in the output are delimited by the null character; if --verbose is also specified then null characters are also used instead of colons and hard tabs:

<source> <NULL> <linenum> <NULL> <pattern> <NULL> <pathname> <NULL>

If -n or --non-matching are specified, non-matching pathnames will also be output, in which case all fields in each output record except for <pathname> will be empty. This can be useful when running non-interactively, so that files can be incrementally streamed to STDIN of a long-running check-ignore process, and for each of these files, STDOUT will indicate whether that file matched a pattern or not. (Without this option, it would be impossible to tell whether the absence of output for a given file meant that it didn't match any pattern, or that the output hadn't been generated yet.)

Buffering happens as documented under the GIT_FLUSH option in **git(1)**. The caller is responsible for avoiding deadlocks caused by overfilling an input buffer or reading from an empty output buffer.

EXIT STATUS

0

One or more of the provided paths is ignored.

1

None of the provided paths are ignored.

128

A fatal error was encountered.

SEE ALSO

gitignore(5) **gitconfig(5)** **git-ls-files(1)**

GIT

Part of the **git(1)** suite