NAME

git-patch-id − Compute unique ID for a patch

SYNOPSIS

git patch−id [−−stable | −−unstable]

DESCRIPTION

Read a patch from the standard input and compute the patch ID for it.

A "patch ID" is nothing but a sum of SHA−1 of the file diffs associated with a patch, with whitespace and line numbers ignored. As such, it's "reasonably stable", but at the same time also reasonably unique, i.e., two patches that have the same "patch ID" are almost guaranteed to be the same thing.

IOW, you can use this thing to look for likely duplicate commits.

When dealing with *git diff−tree* output, it takes advantage of the fact that the patch is prefixed with the object name of the commit, and outputs two 40−byte hexadecimal strings. The first string is the patch ID, and the second string is the commit ID. This can be used to make a mapping from patch ID to commit ID.

OPTIONS

−−stable

Use a "stable" sum of hashes as the patch ID. With this option:

- Reordering file diffs that make up a patch does not affect the ID. In particular, two patches produced by comparing the same two trees with two different settings for "−O<orderfile>" result in the same patch ID signature, thereby allowing the computed result to be used as a key to index some meta−information about the change between the two trees;
- Result is different from the value produced by git 1.9 and older or produced when an "unstable" hash (see −−unstable below) is configured − even when used on a diff output taken without any use of "−O<orderfile>", thereby making existing databases storing such "unstable" or historical patch−ids unusable.

This is the default if patchid.stable is set to true.

−−unstable

Use an "unstable" hash as the patch ID. With this option, the result produced is compatible with the patch−id value produced by git 1.9 and older. Users with pre−existing databases storing patch−ids produced by git 1.9 and older (who do not deal with reordered patches) may want to use this option.

This is the default.

GIT

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