NAME

git-check-mailmap − Show canonical names and email addresses of contacts

SYNOPSIS

git check−mailmap [<options>] <contact>...

DESCRIPTION

For each “Name <user@host>” or “<user@host>” from the command−line or standard input (when using **−−stdin**), look up the person’s canonical name and email address (see "Mapping Authors" below). If found, print them; otherwise print the input as−is.

OPTIONS

−−stdin

Read contacts, one per line, from the standard input after exhausting contacts provided on the command−line.

OUTPUT

For each contact, a single line is output, terminated by a newline. If the name is provided or known to the *mailmap*, “Name <user@host>” is printed; otherwise only “<user@host>” is printed.

MAPPING AUTHORS

If the file **.mailmap** exists at the toplevel of the repository, or at the location pointed to by the mailmap.file or mailmap.blob configuration options, it is used to map author and committer names and email addresses to canonical real names and email addresses.

In the simple form, each line in the file consists of the canonical real name of an author, whitespace, and an email address used in the commit (enclosed by < and >) to map to the name. For example:

Proper Name <commit@email.xx>

The more complex forms are:

<proper@email.xx> <commit@email.xx>

which allows mailmap to replace only the email part of a commit, and:

Proper Name <proper@email.xx> <commit@email.xx>

which allows mailmap to replace both the name and the email of a commit matching the specified commit email address, and:

Proper Name <proper@email.xx> Commit Name <commit@email.xx>

which allows mailmap to replace both the name and the email of a commit matching both the specified commit name and email address.

Example 1: Your history contains commits by two authors, Jane and Joe, whose names appear in the repository under several forms:

Joe Developer <joe@example.com>
Joe R. Developer <joe@example.com>
Jane Doe <jane@example.com>
Jane Doe <jane@laptop.(none)>
Jane D. <jane@desktop.(none)>

Now suppose that Joe wants his middle name initial used, and Jane prefers her family name fully spelled out. A proper **.mailmap** file would look like:

```
Jane Doe         <jane@desktop.(none)>
Joe R. Developer <joe@example.com>
```

Note how there is no need for an entry for **<jane@laptop.(none)>**, because the real name of that author is already correct.

Example 2: Your repository contains commits from the following authors:

```
nick1 <bugs@company.xx>
nick2 <bugs@company.xx>
nick2 <nick2@company.xx>
santa <me@company.xx>
claus <me@company.xx>
CTO <cto@coompany.xx>
```

Then you might want a **.mailmap** file that looks like:

```
<cto@company.xx>                       <cto@coompany.xx>
Some Dude <some@dude.xx>         nick1 <bugs@company.xx>
Other Author <other@author.xx>   nick2 <bugs@company.xx>
Other Author <other@author.xx>         <nick2@company.xx>
Santa Claus <santa.claus@northpole.xx> <me@company.xx>
```

Use hash # for comments that are either on their own line, or after the email address.

GIT

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