

# Separate Git “profiles”

You want to override [Git](#) config (such as commit author email) for only certain repos, such as those under a certain folder. This is useful when dealing with corporate policies, which often block commit pushes that doesn't comfort to certain standards, such as using work email address in the commit email. Those using Bitbucket's [Control Freak](#) may be familiar with this error throw in response `git push`:

```
remote:
remote: Control Freak - Commit 484b773a7e6d2ed8 rejected: bad committer metadata
remote: -----
remote: Committer "John Doe <john.doe@gmail.com>" does not exactly match
remote: a Bitbucket user record. The closest match is:
remote:
remote:      "john.doe <john.doe@somecompany.com>"
```

## Git config has a solution

Git provides a way to solve the above problem – by specifying configuration unique to repos whose paths match a given filepattern. This is achieved using [the `includeIf` section](#) in Git config. But how do we configure this *through* Nix?

## Configuring in home-manager

When using [home-manager](#), you can add the following to your `programs.git` module:

```
programs.git = {
  # Bitbucket git access and policies
  includes = [{
    condition = "gitdir:~/mycompany/**";
    contents = {
      user.email = "john.doe@mycompany.com";
    };
  }];
}
```

With this, any commit you make to repos under the `~/mycompany` directory will use that email address as its commit author email.

# Examples

- [srid/nixos-config: juspay.nix](#) 

