## Accept Developer Certificate of Origin

In order for your contributions to be accepted, you must [sign off](https://git-scm.com/docs/git-commit#git-commit---signoff) your Git commits to indicate that you agree to the terms of [Developer Certificate of Origin](https://developercertificate.org/).

## Follow the Code of Conduct

In order to foster a more inclusive community, Gradle has adopted the [Contributor Covenant](https://www.contributor-covenant.org/version/1/4/code-of-conduct/).

Contributors must follow the Code of Conduct outlined at [https://gradle.org/conduct/](https://gradle.org/conduct/).

## Making Changes

### Development setup

You can test local changes to `gradle-completion.bash` by executing

```bash

source path/to/gradle-completion.bash

```

and then using completion as normal.

For ZSH, set your `fpath` to point to the cloned git repository

```bash

export fpath=($HOME/src/your\_user/gradle-completion \$fpath)

```

Then execute `unfunction \_gradle && autoload -U \_gradle` to reload the gradle completion function, after which you can try completion as normal.

### Debugging

You can get a debug trace of zsh completion by typing `Ctrl-X ?` instead of hitting `[TAB]`. That is especially helpful when submitting bug reports.

Here's a handy [how-to on bash completion](https://iridakos.com/tutorials/2018/03/01/bash-programmable-completion-tutorial.html) and [zsh completion reference](https://github.com/zsh-users/zsh-completions/blob/master/zsh-completions-howto.org).

## Updating Homebrew formula

Updating the `gradle-completion` formula for Homebrew takes extra setup to perform all of the verification Homebrew requires.

First you will need to ensure you can build native ruby gem extensions necessary for `brew audit`:

```

xcode-select --install

brew vendor-install ruby

# Check for any homebrew issues

brew doctor

# Add this to your environment

export PATH="/usr/local/opt/ruby/bin:$PATH"

export LDFLAGS="-L/usr/local/opt/ruby/lib"

export CPPFLAGS="-I/usr/local/opt/ruby/include"

export PKG\_CONFIG\_PATH="/usr/local/opt/ruby/lib/pkgconfig"

```

Ensure you have created a release tag on GitHub for `VERSION`.

```

export GRADLE\_COMPLETION\_VERSION="1.4.1"

wget "https://github.com/gradle/gradle-completion/archive/v${GRADLE\_COMPLETION\_VERSION}.tar.gz"

cat "v${GRADLE\_COMPLETION\_VERSION}.tar.gz" | shasum -a 256 | awk '{print $1}'

# Use brew tools to audit and create PR to homebrew/core — don't forget to replace `VERSION` AND `SHA\_256`!

brew bump-formula-pr --url=https://github.com/gradle/gradle-completion/archive/v<VERSION>.tar.gz --sha256=<SHA\_256> gradle-completion

```

### Creating Commits And Writing Commit Messages

The commit messages that accompany your code changes are an important piece of documentation, please follow these guidelines when writing commit messages:

\* Keep commits discrete: avoid including multiple unrelated changes in a single commit

\* Keep commits self-contained: avoid spreading a single change across multiple commits. A single commit should make sense in isolation

\* If your commit pertains to a GitHub issue, include (`Issue: #123`) in the commit message on a separate line

\* [Sign off](https://git-scm.com/docs/git-commit#git-commit---signoff) your commits to indicate that you agree to the terms of [Developer Certificate of Origin](https://developercertificate.org/).

### Submitting Your Change

After you submit your pull request, a Gradle core developer will review it. It is normal that this takes several iterations, so don't get discouraged by change requests. They ensure the high quality that we all enjoy.

If you need to check on CI status as an external contributor, login as a guest.

### Signing Off Commits After Submitting a Pull Request

Pull requests are automatically verified that all commit messages contain the Signed-off-by line with an email address that matches the commit author. In case you didn't sign off your commits before creating a pull request, you can still fix that to confirm that you agree to the terms of [Developer Certificate of Origin](https://developercertificate.org/).

To sign off a single commit:

`git commit --amend --signoff`

To sign off one or multiple commits:

`git filter-branch --msg-filter "cat - && echo && echo 'Signed-off-by: Your Name <Your.Name@example.com>'" HEAD`

Then force push your branch:

`git push --force origin test-branch`

## Getting Help

If you run into any trouble, please reach out to us on the issue you are working on.

## Our Thanks

We deeply appreciate your effort toward improving Gradle. For any contribution, large or small, you will be immortalized in the release notes for the version you've contributed to.

If you enjoyed this process, perhaps you should consider getting [paid to develop Gradle](https://gradle.com/careers)?