

git config --make-it-work-for-you

\$ whoami



Jordan Duabe

- Software Engineer
- Mechanical keyboard enthusiast
- jordan@jduabe.dev
- @j4ckofalltrades@infosec.exchange



Source: <https://xkcd.com/1597>

Configuring git

git config scopes

system

`$(prefix)/etc/gitconfig`

global

`$XDG_CONFIG_HOME/git/config`
`~/.gitconfig`

local

`$GIT_DIR/config`

"Local wins"



git config priority

Local > Global > System

Quality of life changes

Autocorrect



Enable autocorrect

```
git config --global help.autocorrect prompt
```

```
# git will search for the closest matching valid command and suggest that to you.  
$ git branhc  
WARNING: You called a Git command named 'branhc', which does not exist.  
Run 'branch' instead? (y/N)y  
* main  
  
$ git com  
git: 'com' is not a git command. See 'git --help'.
```

The most similar commands are

commit
column

Aliases



Configure aliases

```
git config --global alias.st "status --short"
```

```
# Alternatively you can edit your `~/.gitconfig` file directly.  
# If you have any external scripts, you can call them from git  
# with `!/path/to/your/script`.
```

[alias]

```
bbr = !better-git-branch.sh  
br = branch  
cm = commit  
di = diff  
dis = diff --staged  
lol = log --oneline  
rs = restore  
rss = restore --staged  
st = status -sb  
sh = stash  
sw = switch
```

```
# Autocomplete also works for configured aliases.  
  
# tab completion works for all aliases  
$ git rs  
rs rss  
  
# if autocorrect is enabled  
$ git cx  
WARNING: You called a Git command named 'cx', which does not exist.  
Run 'cm' instead? (y/N)y  
On branch main  
nothing to commit, working tree clean
```

Sort branches



Sort branches by the most recent commit date

```
git config --global branch.sort -committerdate
```

```
# branches with the most recent commit are listed first
$ git branch
  branch-with-most-recent-commit
  feat-2
  bugfix-1
* main
```

Sort tags



Better tag sorting

git config --global tag.sort version:refname

```
# This setting treats version numbers as integers  
# instead of the full tag name as a string.
```

```
# default "sort" for tags  
$ git tag  
1.0.0  
1.0.1000  
1.0.101  
1.0.2
```

```
# when tag.sort is set to version:refname  
$ git tag  
1.0.0  
1.0.2  
1.0.101  
1.0.1000
```

Fetch



Better fetch

```
git config --global fetch.prune true
```

```
git config --global fetch.pruneTags true
```

Push



Better push

```
git config --global push.default simple # default since 2.0
```

```
git config --global push.autoSetupRemote true
```

```
git config --global push.followTags true
```

```
# `push.autoSetupRemote = true` gets rid of the following error  
# which you might be familiar with:
```

```
$ git push  
fatal: The current branch my-branch-name has no upstream branch.  
To push the current branch and set the remote as upstream, use  
  
git push --set-upstream origin my-branch-name
```



Better diff

```
git config --global diff.colorMoved plain
```

```
git config --global diff.mnemonicPrefix true
```

```
git config --global diff.renames true
```

```
git config --global diff.algorithm histogram
```

mnemonic prefix

```
$ git diff  
diff --git a/git-config/index.html b/git-config/index.html  
index 98ede1b..884b1f1 100644  
--- a/git-config/index.html  
+++ b/git-config/index.html
```

```
# diff.mnemonicPrefix true  
$ git diff  
diff --git i/git-config/index.html w/git-config/index.html  
index 98ede1b..884b1f1 100644  
--- i/git-config/index.html  
+++ w/git-config/index.html
```

myers vs histogram algorithm

```
-- myers
void func1() {
    x += 1
}
+
+void functhreehalves() {
+    x += 1.5
}

-- histogram
void func1() {
    x += 1
}

+void functhreehalves() {
+    x += 1.5
}
+
```



Configure git to use the best text editor

```
git config --global core.editor vim
```

Commit with diffs



Show diff in the commit editor

```
git config --global commit.verbose true
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#   modified: README.md
#
# ----- >8 -----
# Do not modify or remove the line above.
# Everything below it will be ignored.
diff --git c/README.md i/README.md
index d6166fe..71ee90f 100644
--- c/README.md
+++ i/README.md
```

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSOKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

Source: <https://xkcd.com/1296/>

Global gitignore



Setup a global .gitignore for commonly ignored files

```
git config --global core.excludesfile ~/.gitignore-global
```

```
.DS_Store
```

```
*.orig
```

```
*.swp
```

```
.idea/
```

```
.vscode/
```

"It depends" / Matter of preference

External tooling: delta



Use delta as a pager

```
git config --global core.pager delta
```

```
git config --global interactive.diffFilter 'delta --color-only'
```

delta side-by-side diff

Δ [content/posts/2025/git-config.md](#)

• 1:

```
1 ---  
2 title: git config  
3 date: 2025-03-15T12:21:19+08:00  
4 description:  
5 draft: true  
6 categories:  
7 tags:  
8 series:  
9 images:  
10 featured_image:  
11 asciinema: true
```

```
1 ---  
2 title: What's in my .gitconfig?  
3 date: 2025-03-15T12:21:19+08:00  
4 description: Diving into my current git configuration.  
5 draft: false  
6 categories:  
7 tags:  
8 series:  
9 images:  
10 featured_image:  
11 asciinema: true
```

What is your merge strategy?

Source: <https://giphy.com/gifs/git-merge-cFkiFMDg3iFol>

What is your merge strategy?



Source: <https://giphy.com/gifs/git-merge-cFkiFMDg3iFol>

Rebase



Quality of life changes if you use *rebase* as your merge strategy

```
git config --global pull.rebase true
```

```
git config --global rebase.autoSquash true
```

```
git config --global rebase.autoStash true
```

```
git config --global rebase.updateRefs true
```

Reuse Recorded Resolution



Don't Repeat Yourself (DRY) for resolving conflicts

```
git config --global rerere.enabled true
```

```
git config --global rerere.autoUpdate true
```

Signing commits and tags



Configure git to sign commits and tags with a GPG key

```
git config --global user.signingKey XXXX # what key to use
```

```
git config --global commit.gpgSign true # sign commits
```

```
git config --global tag.gpgSign true # sign tags
```



Configure git to sign commits and tags with a SSH key

```
git config --global gpg.format ssh
```

```
git config --global user.signingkey /path/to/.ssh/key.pub
```

Git forges e.g. GitHub will then mark signed commits (or tags) that are verifiable as *Verified*.

```
$ git log --show-signature
commit 9627416a5acb42eef9eac79d2b59765f6ca39510 (HEAD -> main)
gpg: Signature made Sat 30 Aug 2025 06:49:36 PM AWST
gpg:           using EDDSA key A8C6DD14ECFC45C04F06535149CC7B953A9FA12D
gpg: Good signature from "Jordan Duabe <me@jduabe.dev>" [ultimate]
Author: Jordan Duabe <me@jduabe.dev>
Date:  Sat Aug 30 18:49:19 2025 +0800
```

Update trusted publishing & attestations talk

Add downloads badge

 j4ckofalltrades committed on Oct 27, 2024 · ✓ 1 / 1

0b29acf  

Verified

Conditional includes

```
$ tree
.
├── .gitconfig
├── .gitconfig-oss
├── .gitconfig-work
└── oss
    └── oss-repo-1
└── work
    └── work-repo-1

$ cat .gitconfig
[includeIf "gitdir:~/oss/"]
  path = ~/.gitconfig-oss

[includeIf "gitdir:~/work/"]
  path = ~/.gitconfig-work

$ cat .gitconfig-oss
[user]
  email = my-oss-email.com

$ cat .gitconfig-work
[user]
  email = my-work-email.com
  signingKey = 101XYZ101
```

Takeaways

- Experiment and see what fits with your workflow.
- The best configuration is the one that works for you.

Recommended reading

- [Pro Git Book](#)
- [Julia Evans' blog posts](#)
- [So You Think You Know Git?](#)
- [Jujutsu](#)
- [What's in my .gitconfig?](#)

Links

L: Link to gist for git config, R: Link to slides



Thank You



- ⦿ 1. git commit
- ⬆ 2. git push
- ➡ 3. leave building

Source: https://raw.githubusercontent.com/hendrixroa/in-case-of-fire-1/master/in_case_of_fire.png