

Advanced



Created by Guy Van den Nieuwenhof

<https://github.com/guyvdn>

Agenda

- Properly configure Git
- Advanced Git commands
- Customize Git with Git Hooks
- Further Reading

Agenda

- **Properly configure Git**
- Advanced Git commands
- Customize Git with Git Hooks
- Further Reading

Properly configure Git

1. Files
2. Your Identity
3. Git Credential Manager
4. Editors
5. Checking Your Settings

Files

1. **/etc/gitconfig**

System-wide configuration file.

Pass the option **--system** to git config.

2. **~/.gitconfig** or **~/.config/git/config**

User-specific configuration file.

Pass the option **--global** to git config.

3. **.git/config** (default)

Repository specific configuration file.

Pass the option **--local** to git config.

Your Identity

Global Identity

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@gmail.com
```

Identity for customer specific repository

```
$ git config --local user.email johndoe@customer.com
```

Git Credential Manager

Git Credential Managers simplify authentication with your Git repos. They also provide support two-factor authentication.

Windows

Download and run the latest Git for Windows installer, which includes the Git Credential Manager for Windows.

<https://git-scm.com/download/win>

macOS and Linux

```
> git-credential-manager install
```

Editors

Text Editor

```
$ git config --global core.editor %  
"'C:/Program Files (x86)/Notepad++/notepad++.exe' %  
-multiInst -notabbar -nosession"
```

Merge and Diff Tool

```
$ git config --global --add merge.tool kdiff3  
$ git config --global --add mergetool.kdiff3.path "C:/../kdiff3.exe"  
$ git config --global --add mergetool.kdiff3.trustExitCode false  
  
$ git config --global --add diff.guitool kdiff3  
$ git config --global --add difftool.kdiff3.path "C:/../kdiff3.exe"  
$ git config --global --add difftool.kdiff3.trustExitCode false
```


Checking Your Settings

If you want to check your configuration settings, you can use the `git config --list` command to list all the settings Git can find at that point.

```
$ git config --list
user.name=John Doe
user.email=johndoe@example.com
..
```

You may see keys more than once, because Git reads the same key from different files (`~/.gitconfig` and `.git/config`, for example). In this case, Git uses the last value for each unique key it sees.

Checking Your Settings

You can also check what Git thinks a specific key's value is by typing `git config <key>`

```
$ git config user.name  
John Doe
```

Agenda

- Properly configure Git
- **Advanced Git commands**
- Customize Git with Git Hooks
- Further Reading

Agenda

- Properly configure Git
- Advanced Git commands
- Customize Git with Git Hooks
- **Further Reading**

Further Reading

Git Reference Manual

<https://git-scm.com/docs>

The official and comprehensive man pages that are included in the Git package itself

GitHub Cheat Sheet

<https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>

Summarizes commonly used Git command line instructions for quick reference

Visual Git Cheat Sheet

<http://ndpsoftware.com/git-cheatsheet.html>

An interactive cheat sheet from NDP Software

Pro Git book

<https://git-scm.com/book>

Free book by **Scott Chacon** and **Ben Straub**. Also available on [amazon.com](https://www.amazon.com)