Chapter 2: Initial .gitconfig setup

By Ryan Peters

Table of Contents

[Chapter 2: Initial .gitconfig setup 6](#_Toc499485758)

[Utilizing the Command line 6](#_Toc499485759)

[Choosing Command Scope 6](#_Toc499485760)

[Strategy for understanding limitations in scope 7](#_Toc499485761)

[Using the Core Commands 7](#_Toc499485762)

[git config --global user.name “your name” 7](#_Toc499485763)

[git config --global user.email “yourEmail@provider.suffix” 7](#_Toc499485764)

[git config --global core.editor " ‘d:/’path to’/your\_chosen/editor.exe’ -arg" 7](#_Toc499485765)

[References 8](#_Toc499485766)

Proper initialization of git on a new workstation will require you to perform an initial set-up on the git config files affecting your workspace. This chapter covers how to set up the core config file components for basic use of git.

# Utilizing the Command line

Professional software engineers prefer using git from the command line as it permits flexibility and customization of functionality. Commands for config settings shall be presented in the following command terminal format:

git config [<scope choice>] <command>

The square brackets [ ] denote optional arguments, and the contents of the diamond brackets < > denote what type of argument to use, as shown in *Table 1*.

# Choosing Command Scope

It is important to know when to use the --system, --global, and --local specifiers for a given command. These specifiers are used to tell git how wide the scope of influence should be for the command that follows it.

--system This scope is applied to all repositories on the workstation, regardless of which user is currently logged in.

--global This scope is applied to the repositories which fall under subdirectories of the user’s profile directory.

--local This scope is applied to repositories under the current folder and its subdirectories’ level of scope.

## Strategy for understanding limitations in scope

It can often be difficult to keep a clear sense of which scope specifier will affect a given repository, so it helps to think of the reach for a scope-specifier as it relates to the distance between your repository and where the config file is saved.

### Config File Locations

Git will create and maintain up to 4 different config files, which are saved in locations prescribed according to their intended scope of influence:

1. Your workstation’s system wide .gitconfig  file; the specific location will depend on your operating system.
2. Your user profile’s .gitconfig file located at the following 2 locations:
   1. ~/.gitconfig.
   2. ~/.config/git/config.
3. The local repository’s config file: ./.git/config.

### Config File precedence

The config files defined in bullet points 1, 2, and 3, above, are used to build a cascading list of settings for your local repository in the following order of precedence:

1. System config.
2. Global (user) config.
3. Local (repository-specific) config.

Each successive layer represents a more localized config file, the last and most local file takes greatest precedence.

# Using the Core Commands

Though the config file can contain many settings for use in specific tasks, there are 3 which should generally be set to the global scope: user.name, user.email, and core.editor.

## git config --global user.name “your name”

Setting user.name will allow git to automatically add your name to any commit, tag, or push you perform. This greatly improves team communication and understanding when tracing out bugs. It also creates a written record of authorship.

## git config --global user.email “yourEmail@provider.suffix”

Setting user.email will allow git to automatically include your email address in any tag, commit, or push you perform.

## git config --global core.editor " ‘d:/’path to’/your\_chosen/editor.exe’ -arg"

Setting core.editor tells git to no longer use the default text editor, vim, and instead use the editor at the file path specified inside of the set of single quotation marks.

#### Table 1: Core .gitconfig commands

|  |  |  |
| --- | --- | --- |
| **Commands** <command> | **Applicable Scopes** <scope choice> | **Syntactic Notes** |
| **user.name** | --global  --local | For git to understand that your first and last name should be treated as a single string, you need to surround your whole name in single or double quotation marks. |
| **user.email** | --global  --local | In order to prevent possible conflicts with your email containing special characters, surround your full email address in single or double quotation marks. |
| **core.editor** | --system  --global | Inside the double quotation mark, but outside the single quotation mark, you may include command line arguments which will modify the conditions under which the application opens. |

# References

*Git - First-Time Git Setup*. https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup. Accessed 26 Nov. 2017.

*Git - Git-Config Documentation*. https://git-scm.com/docs/git-config. Accessed 26 Nov. 2017.

*The Three Git Config Files*. http://www.whiteboardcoder.com/2014/08/the-three-git-config-files.html. Accessed 26 Nov. 2017.

**[word count 600]**