Chapter 2: Initial .gitconfig setup

By Ryan Peters

Table of Contents

[Utilizing the Command line 6](#_Toc499554202)

[Choosing Command Scope 6](#_Toc499554203)

[Strategy for understanding limitations in scope 7](#_Toc499554204)

[Using the Core Commands 7](#_Toc499554205)

[git config --global user.name “your name” 7](#_Toc499554206)

[git config --global user.email yourEmail@provider.suffix 7](#_Toc499554207)

[git config --global core.editor "‘d:/path to/your\_chosen/editor.exe’ [–cmd\_line\_arg]" 7](#_Toc499554208)

[Table 1: Core .gitconfig commands 8](#_Toc499554209)

[References 8](#_Toc499554210)

Proper initialization of git on a new workstation will require you to perform an initial set-up on the .gitconfig (aka ‘git config’) files affecting your workspace. This chapter covers how to set up the core components of the git config files 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 git config settings will 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, as shown in *Table 1*, to use.

# Choosing Command Scope

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

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

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

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

## Strategy for understanding limitations in scope

It helps to think of the reach of a scope-specifier as it relates to the distance between your repository and where the config file is saved. Git uses these 4 directory locations:

1. Your machine's system config file, the specific location will depend on your operating system.
2. Your user profile’s config file located at one of the following 2 locations:
   * ~/.gitconfig
   * ~/.config/git/config
3. The local repository’s config file:
   * ./.git/config

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 .gitconfig 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. For further details concerning what scope specifiers are appropriate for each of the core commands, as well as notes about specific syntax, see Table 1.

## 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’ [–cmd\_line\_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 | Note the use of quotation marks around the user’s whole name. Using quotation marks like this tells git to treat first and last as a single string. |
| **user.email** | --global  --local |  |
| **core.editor** | --system  --global | You can add command line arguments if you wish inside the double quotes. |
| *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. | | |

# 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 587]**