Aa

A

Appendix A: Installing & Configuring Git Written by Chris Belanger & Bhagat Singh

Installing Git is relatively straightforward, but putting a little care in your initial setup and configuration will go a long way to ensuring that your work with Git is as hassle-free as possible.

Installing on Windows

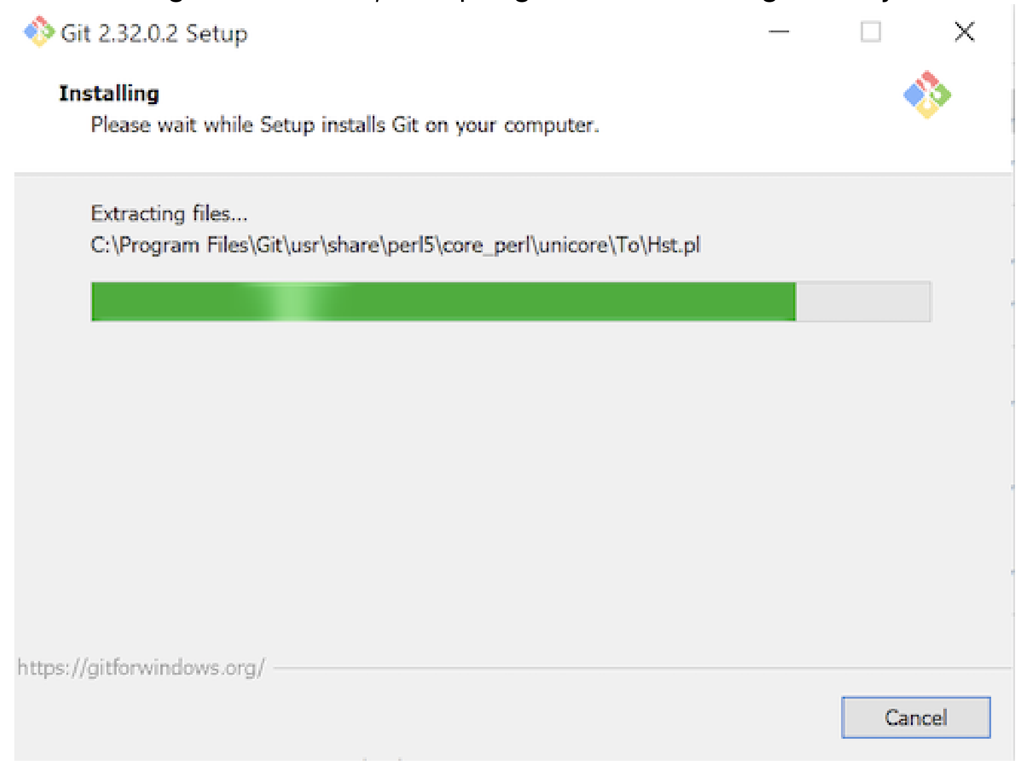
To remain as platform-agnostic as possible, you'll install Git using one of the official standalone installers. While you can use the Chocolatey Package Manager for Windows, or even download and install GitHub Desktop (which installs Git on its own), you'll install and configure the plain-vanilla version of Git for Windows.
These instructions were tested on Windows 10, but the concepts should be similar across Windows versions.

. Download the official release of Git for Windows at the following link:
https://github.com/git-for-windows/git/releases/
This book uses the 2.32.0 release, available here, but anything above version 2.32 should work just fine:
https://github.com/git-for-windows/git/releases/tag/v2.32.0.windows.2

. Execute the self-contained EXE file once it downloads fully.

. If prompted to allow changes to your system, click **OK**.

. Click through the installer, accepting all defaults along the way. One thing you might want to change, depending on your system, is the install location of Git, which by default is **C:\Program Files\Git**. If you usually install everything into **C:\Program Files**, then you can leave this option alone.



Git 2.32.0.2 Setup

Installing
Please wait while Setup installs Git on your computer.

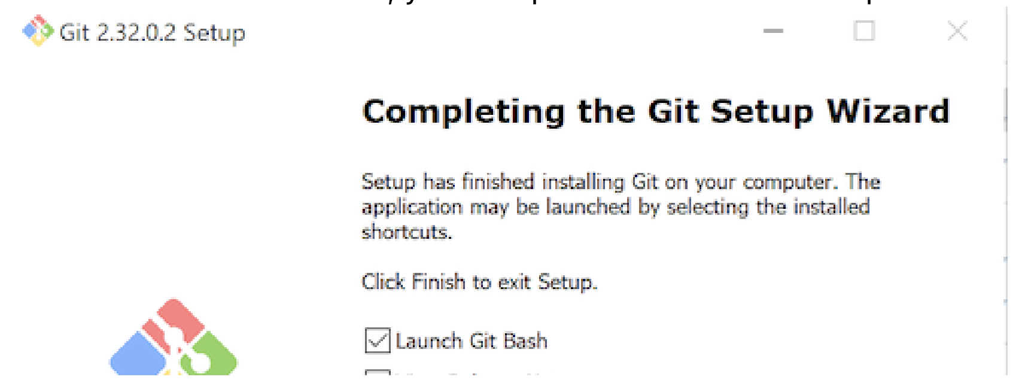
Extracting files...
C:\Program Files\Git\usr\share\perl5\core_perl\unicore\To\Hst.pl

https://gitforwindows.org/

Note: To best follow along with this book, leave the default editor option as **Vim**.
Although you can select from a list of arguably excellent and more user-friendly editors as part of the setup process, you'll likely get lost when you try to use another text editor to create your commit messages or do other tasks. But if you feel compelled to choose another option, do so now. *Dulce periculum*. :]

. The app will install Git and a host of helper libraries. This will only take a Microsoft minute.

. When the installer finishes, you'll be presented with the completion dialog. Unselect **View Release Notes** (you have this book, so who needs release notes anyway?) and select **Launch Git Bash** so you can start the configuration process once the installer closes. Then click **Next**.



Git 2.32.0.2 Setup

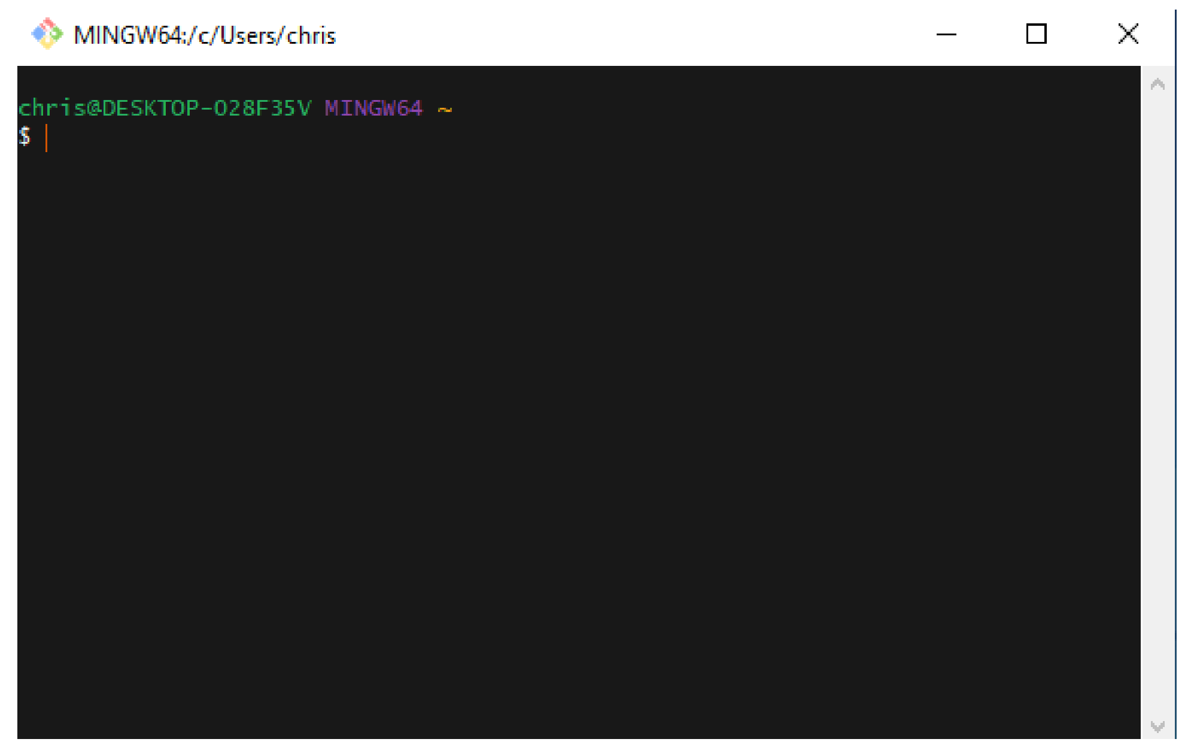
Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch Git Bash

. You'll see a console that looks similar to the following:



```
chris@DESKTOP-028F35V MINGW64 ~
$ 
```

This is **Git Bash**. It's similar to the familiar Windows Command shell, but it's a version of the Bourne Again Shell (**bash**) that's a common way for people to interact with Linux, macOS and other platforms.
If you use the Git Bash shell to interact with your directories and projects, you'll be able to follow along with this book pretty much verbatim. This includes using the command line tools in this book, such as `nano`.
However, if you choose to use Git CMD (which lets you use the familiar Windows path structure, among other things), you'll have to adapt some of the commands and/or tools that you'll use in this book to their Windows equivalents.

Installing on macOS

There are a few ways to install Git on macOS. There *is* a standalone installer available for Git, which installs Git through a GUI interface. Also, installing GitHub Desktop will automatically set up Git for you. However, the two recommended methods for maximum control are to either install with Xcode's command-line tools or use the Homebrew package manager to install Git on your system. Iterating again, that the book uses version 2.32.0, but anything above that should be just fine.

Installing Xcode's command-line tools

Chances are you're using Xcode if you're developing on a Mac. Since Xcode has some really good Git integration, you might as well just let Xcode do what it wants and manage the Git installation itself.

. If you have Xcode installed on your Mac, simply execute the following command from Terminal to install the Xcode command-line tools:

```
xcode-select --install
```

You'll see a prompt to install the Xcode command-line tools, which include Git. Simply wait for the installer to run and finish.

. Run the following command to verify which version of Git installed:

```
git --version
```

If Git responds with the current version or greater, which is 2.32.0 as of this writing, then you're good. If the version number is older, verify that you have the most recent version of Xcode installed.

Installing with Homebrew

Homebrew is a useful package manager for macOS. With Homebrew, you can install and update hundreds, if not thousands, of pieces of software right from your command line.
To install Homebrew, execute the following command in Terminal:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Sit back and wait while Homebrew installs itself. This might take a while.
Once Homebrew has finished installing, you can get down to installing Git.

. Execute the following command in Terminal to install Git using Homebrew:

```
brew install git
```

Once again, sit back and wait for Homebrew to do its thing.

. Once Homebrew has finished installing Git, you'll need to check that Homebrew has actually been able to overwrite any existing installations of Git on your machine.
To check which version of Git is being pointed to on your system, execute the following command in Terminal:

```
git --version
```

If Git responds with the current version, which is 2.32.0 as of this writing, then you're good. You're still fine if it is greater than 2.32.
If Git responds with an older version, then the symlinks that point to the Homebrew-installed copy of Git haven't been updated properly. Force Homebrew to rebuild those symlinks with the following command:

```
brew link --overwrite git
```

Execute `git --version` once more and Git should now report the correct version.

Configuring credentials

There are a few things you can do once you've installed Git to make your life a tiny bit easier; they're optional but highly recommended. One of those things is to set up your GitHub credentials in Git so they stick, saving you from having to re-enter them frequently.

Setting your username and email

. To persist your GitHub username so you don't have to type it in every time you push your changes to a remote repository, execute the following command, enclosing your name in quotes:

```
git config --global user.name "your-username-here"
```

. To persist the email you use for commits, which will appear alongside your commit history, enter the following command, enclosing your email in quotes:

```
git config --global user.email "youremail@domain.com"
```

Persisting your password

If you're on macOS, authenticating against GitHub or other repositories from the command line will store your password on the macOS Keychain, so you won't have to enter your credentials each time you want to interact with a remote repository.
If you're on Windows, you'll need to install the Git Credential Manager for Windows to get the same functionality. Find the instructions for installing that helper tool here:
https://github.com/Microsoft/Git-Credential-Manager-for-Windows

11. Conclusion

Have a technical question? Want to report a bug? You can ask questions and report bugs to the book authors in our official book forum here.