A Little Rails

Creating a stable, useful development environment is one of the most difficult tasks in computing. We are forever chasing our tails: new versions of languages, frameworks, tools, operating systems, etc. take extraordinary amounts of developer time - time that could otherwise be spent actually developing software. It is also incredibly frustrating. What works for one person doesn't for another, and what works one day doesn't work the next. Every operating system, language, and framework has its own method of managing their associated packages. It's all a huge drain on the software industry, and indeed on everyone who uses computers. It's no wonder how reticent everyone is to upgrade their devices. All that said, we must endeavor to create such development environments or be unable to develop.

For Ruby on Rails, you first install a Ruby environment, and use that to install Rails. This chapter will cover installing both on Windows, OSX, Linux, and using two platform-independent approaches.

Install Ruby

Windows

There are two major approaches to create a RoR environment on Windows: use the Windows Subsystem for Linux (WSL) and use the Linux approach to Ruby and Rails, or download a Ruby installer.

WSL

First, install a Linux version using WSL. Bring up PowerShell and enter

```
> wsl --install
```

You can choose which Linux distribution to install; the default is Ubuntu. To see the list of available distributions, enter

```
> wsl --list --online
```

You can install multiple distributions if you want. Now bring up a terminal to use the Linux package manager to see if the correct version of Ruby is available. The following command

```
> apt search ruby
```

will show the various (and typically plentiful!) packages related to Ruby. Here, we're looking for a version that starts with 3. If there isn't one directly available via the apt manager, we have to use a different approach.

rbenv

There are several version managers for Ruby, this book recommends rbenv (https://github.com/rbenv/rbenv). rbenv is typically available via Linux package managers, so the following will install it on Debian/Ubuntu systems:

```
> sudo apt install rbenv
```

The "sudo" command allows you to elevate your permissions for the one command (to the "root" user) in order to write files to special places in the operating system. You should use sudo with great care as it can do a great deal of damage if used recklessly. You can then set up rbenv via

```
> rbenv init

# Load rbenv automatically by appending

# the following to ~/.bash_profile:



eval "$(rbenv init - bash)"
```

Do as it says and add that line at the end of your ~/.bash_profile file so that rbenv works every time you log in. rbenv creates a directory in your home ('~') directory named ".rbenv". In there, there are two important directories: 'shims' and 'versions'. In order for rbenv to work, the '~/.rbenv/shims' directory must occur early in your $PATH environment variable. For example:

```
> echo $PATH

/home/beaty/.rbenv/shims:/home/beaty/bin:/usr/local/sbin:/usr/local/bin:/u
sr/sbin:/usr/bin:/sbin:/bin
```

shows that the shims directory is before other possible places a ruby executable might reside. To find the stable versions of Ruby, do:

```
> rbenv install -l

2.6.9

2.7.5

3.0.3

3.1.1

jruby-9.3.4.0

mruby-3.0.0

rbx-5.0

truffleruby-22.0.0.2

truffleruby+graalvm-22.0.0.2
```

If you don't see many or any, you might need to update rbenv's plugin directory for ruby-build via:

```
> mkdir -p "$(rbenv root)"/plugins

>    git    clone    https://github.com/rbenv/ruby-build.git    "$(rbenv
root)"/plugins/ruby-build
```

Now you should be able to install a Ruby version that starts with 3:

```
> rbenv install 3.1.1
```

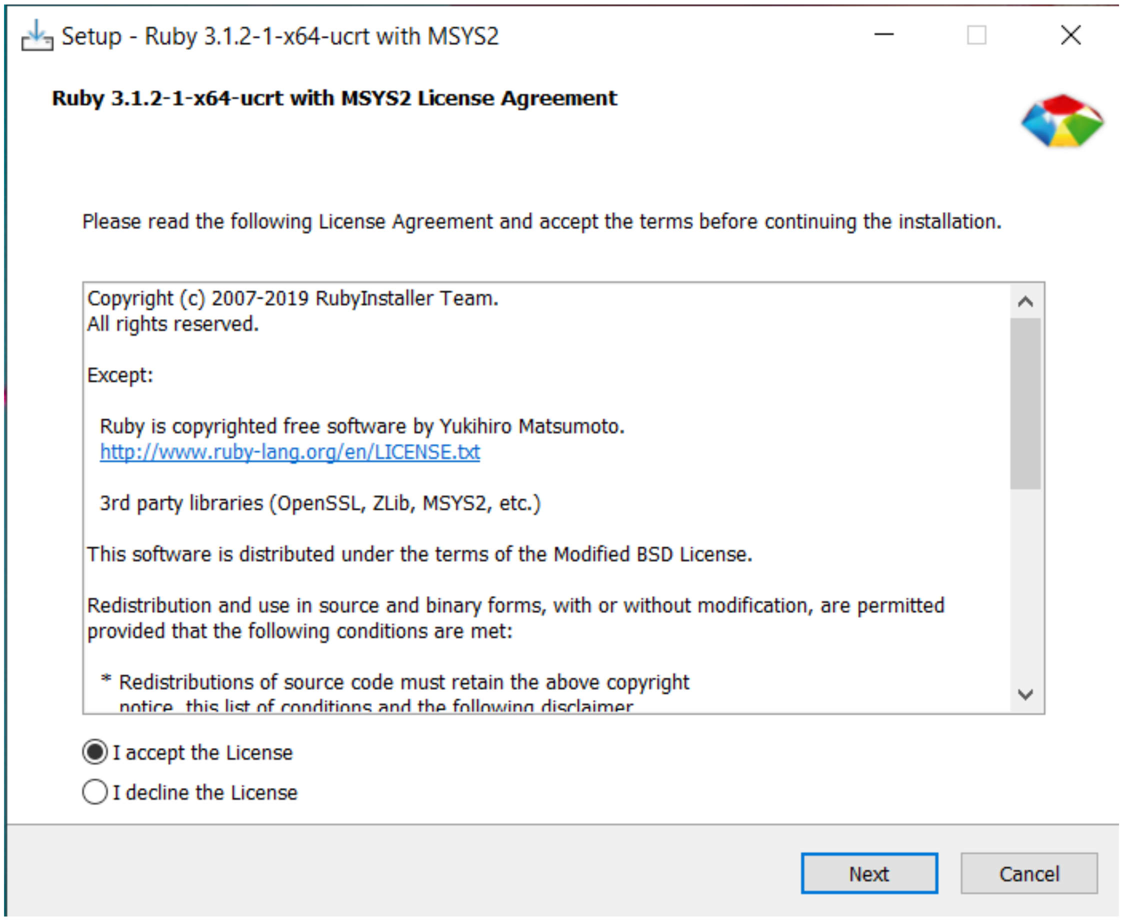
```
Downloading openssl-1.1.1n.tar.gz...

->
https://dqw8nmjcqpjn7.cloudfront.net/40dceb51a4f6a5275bde0e6bf20ef4b91bfc3
2ed57c0552e2e8e15463372b17a

Installing openssl-1.1.1n...

Installed openssl-1.1.1n to /Users/stevebeaty/.rbenv/versions/3.1.1



Downloading ruby-3.1.1.tar.gz...

-> https://cache.ruby-lang.org/pub/ruby/3.1/ruby-3.1.1.tar.gz

Installing ruby-3.1.1...

ruby-build: using readline from homebrew

Installed ruby-3.1.1 to /Users/stevebeaty/.rbenv/versions/3.1.1
```

You can now check which Ruby you're running and the version number:

```
> which ruby

/Users/stevebeaty/.rbenv/shims/ruby

> ruby --version

ruby 3.1.1p18 (2022-02-18 revision 53f5fc4236) [x86_64-darwin20]
```

Ruby Installer

Go to https://rubyinstaller.org/downloads/ and choose the recommended version with the devkit -- currently Ruby+Devkit 3.1.2-1 (x64) for example. This will install everything you need for Ruby and you can manage it as you do with other software packages you install. There are a number of screens that will come up during installation, though happily the defaults are reasonable. Here's the first where you must accept the license after having read and understood it.



Ruby 3.1.2-1-x64-ucrt with MSYS2 License Agreement

Please read the following License Agreement and accept the terms before continuing the installation.

Copyright (c) 2007-2019 RubyInstaller Team.
All rights reserved.

Except:

Ruby is copyrighted free software by Yukihiro Matsumoto.
http://www.ruby-lang.org/en/LICENSE.txt

3rd party libraries (OpenSSL, ZLib, MSYS2, etc.)

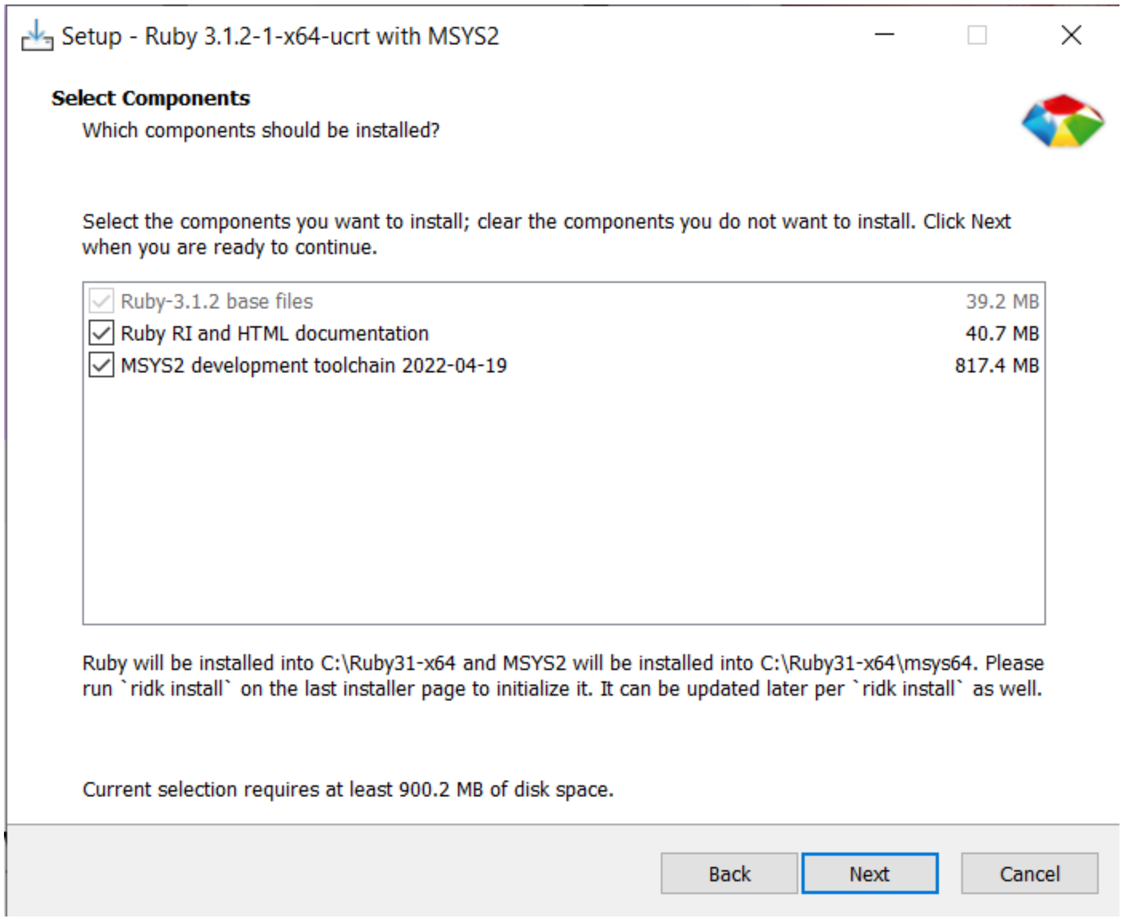
This software is distributed under the terms of the Modified BSD License.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer

[✓ I accept the License]
I decline the License

The next asks what you want to install, the defaults (everything) is what you want.


Setup - Ruby 3.1.2-1-x64-ucrt with MSYS2

Select Components

Which components should be installed?

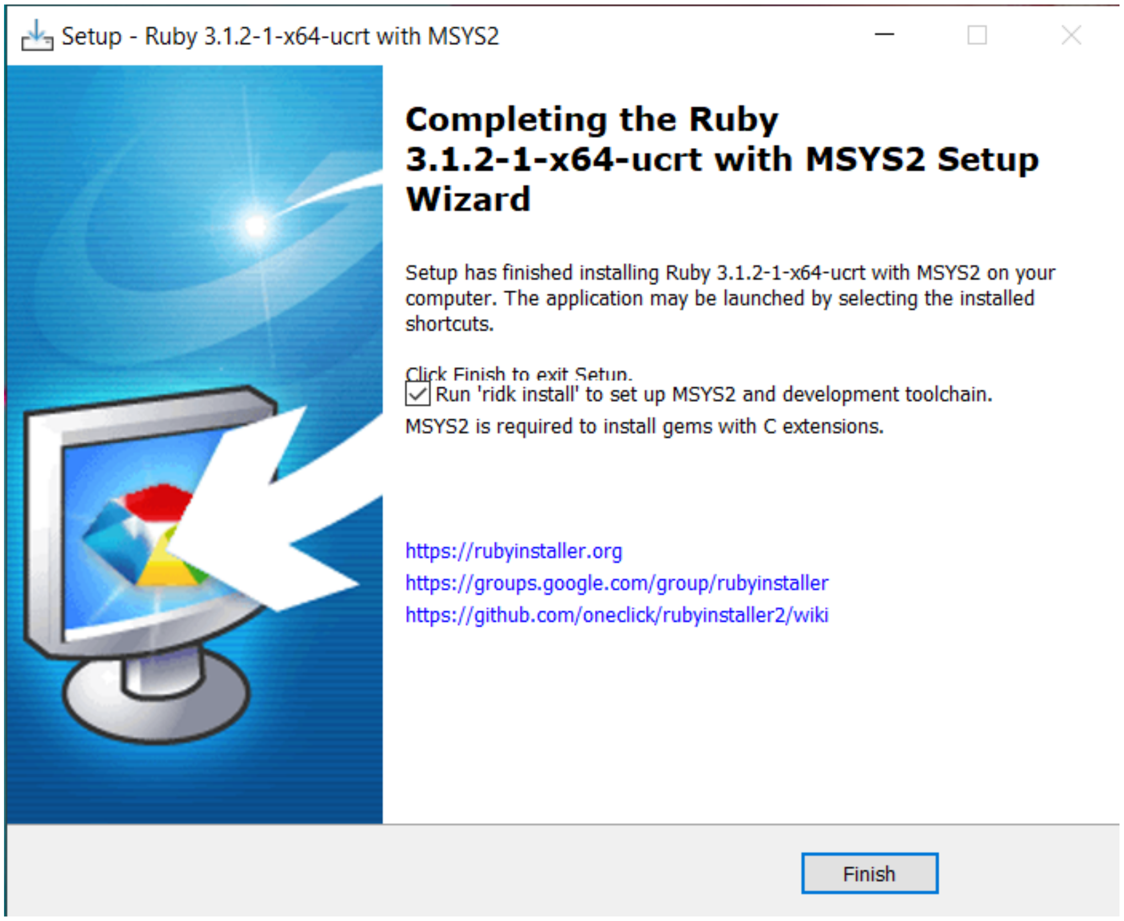
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

✓ Ruby-3.1.2 base files	39.2 MB
✓ Ruby RI and HTML documentation	40.7 MB
✓ MSYS2 development toolchain 2022-04-19	817.4 MB

Ruby will be installed into C:\Ruby31-x64 and MSYS2 will be installed into C:\Ruby31-x64\msys64. Please run `ridk install` on the last installer page to initialize it. It can be updated later per `ridk install` as well.

Current selection requires at least 900.2 MB of disk space.

After that, a screen comes up asking if you want to run “ridk install:, you do.


Setup - Ruby 3.1.2-1-x64-ucrt with MSYS2

Completing the Ruby 3.1.2-1-x64-ucrt with MSYS2 Setup Wizard

Setup has finished installing Ruby 3.1.2-1-x64-ucrt with MSYS2 on your computer. The application may be launched by selecting the installed shortcuts.

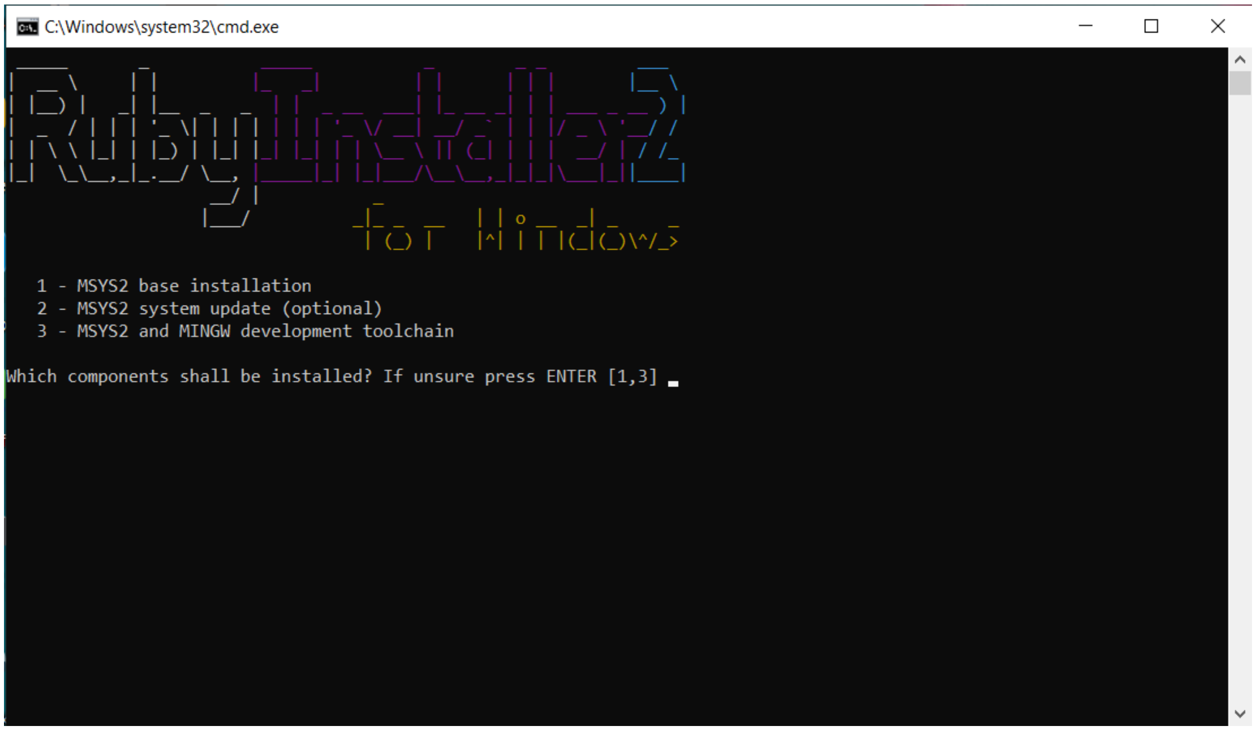
Click Finish to exit Setup.

[✓] Run 'ridk install' to set up MSYS2 and development toolchain.

MSYS2 is required to install gems with C extensions.

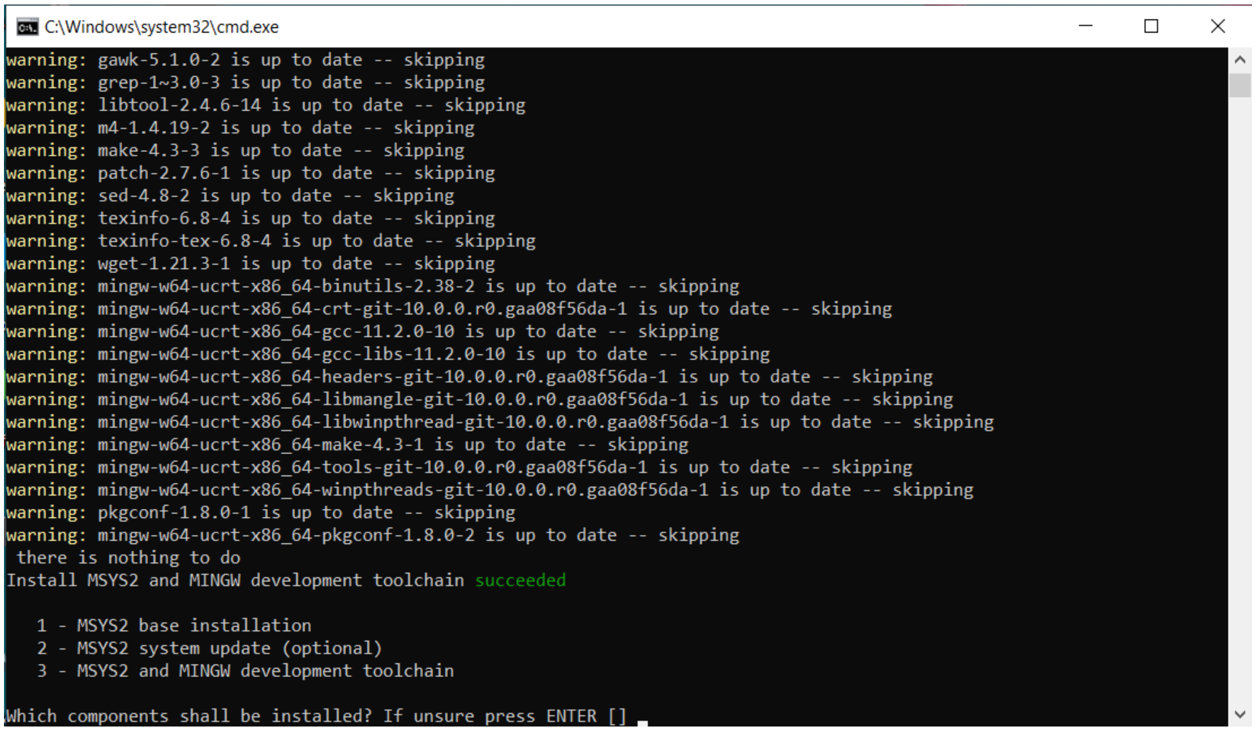
https://rubyinstaller.org
https://groups.google.com/group/rubyinstaller
https://github.com/oneclick/rubyinstaller2/wiki

A "cmd.exe" windows appears asking for what you want installed; again the defaults are appropriate.


C:\Windows\system32\cmd.exe

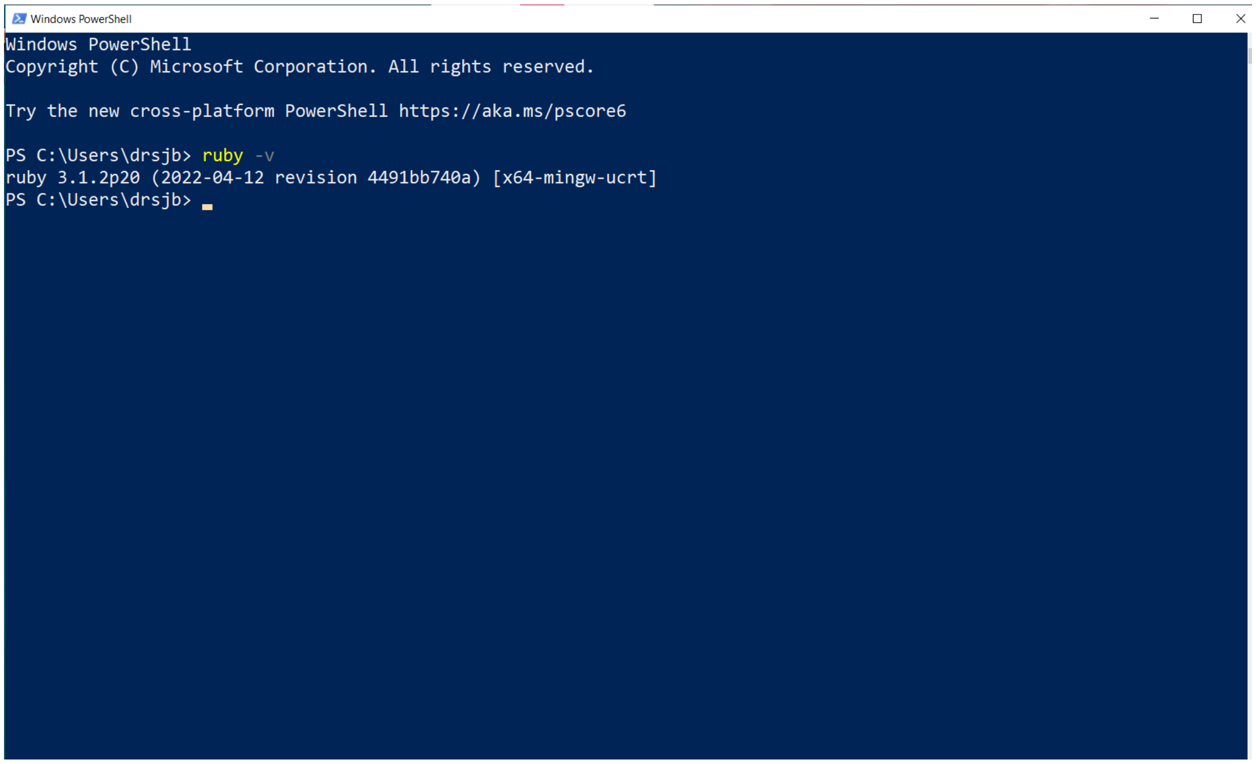
```
RubyInstaller2
for Windows

   1 - MSYS2 base installation
   2 - MSYS2 system update (optional)
   3 - MSYS2 and MINGW development toolchain

Which components shall be installed? If unsure press ENTER [1,3]
```

At the end of that, you will be prompted for what to do next, hitting Enter is appropriate.



```
warning: gawk-5.1.0-2 is up to date -- skipping
warning: grep-1~3.0-3 is up to date -- skipping
warning: libtool-2.4.6-14 is up to date -- skipping
warning: m4-1.4.19-2 is up to date -- skipping
warning: make-4.3-3 is up to date -- skipping
warning: patch-2.7.6-1 is up to date -- skipping
warning: sed-4.8-2 is up to date -- skipping
warning: texinfo-6.8-4 is up to date -- skipping
warning: texinfo-tex-6.8-4 is up to date -- skipping
warning: wget-1.21.3-1 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-binutils-2.38-2 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-crt-git-10.0.0.r0.gaa08f56da-1 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-gcc-11.2.0-10 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-gcc-libs-11.2.0-10 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-headers-git-10.0.0.r0.gaa08f56da-1 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-libmangle-git-10.0.0.r0.gaa08f56da-1 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-libwinpthread-git-10.0.0.r0.gaa08f56da-1 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-make-4.3-1 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-tools-git-10.0.0.r0.gaa08f56da-1 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-winpthreads-git-10.0.0.r0.gaa08f56da-1 is up to date -- skipping
warning: pkgconf-1.8.0-1 is up to date -- skipping
warning: mingw-w64-ucrt-x86_64-pkgconf-1.8.0-2 is up to date -- skipping
 there is nothing to do
Install MSYS2 and MINGW development toolchain succeeded

   1 - MSYS2 base installation
   2 - MSYS2 system update (optional)
   3 - MSYS2 and MINGW development toolchain

Which components shall be installed? If unsure press ENTER []
```

You can now bring up a PowerShell window and make sure you have a working Ruby.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\drsjb> ruby -v
ruby 3.1.2p20 (2022-04-12 revision 4491bb740a) [x64-mingw-ucrt]
PS C:\Users\drsjb>
```

OSX

Macs come with Ruby already installed, but it's not the Ruby you want:

```
> which ruby

/usr/bin/ruby

> /usr/bin/ruby --version

ruby 2.6.3p62 (2019-04-16 revision 67580) [universal.x86_64-darwin20]
```

OSX uses Ruby for various system administration tasks and you don't want to break any of those. Use Homebrew (https://brew.sh/) to install a software manager for packages you need. It relatively easy to use the instructions at that page. You'll first need to install the XCode command-line utilities:

```
> xcode-select –install
```

and then

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

You will be prompted for your password so the brew directories can have the correct permissions set. It's recommended that you run

```
> brew doctor
```

to check that everything installed correctly. You can then search for the various versions of Ruby Homebrew supports:

```
> brew search ruby

==> Formulae

chruby                              ruby-completion

chruby-fish                         ruby-install

cucumber-ruby                       ruby@2.4

imessage-ruby                       ruby@2.5

jruby                               ruby@2.6

mruby                               ruby@2.7

mruby-cli                           ruby@3.0

rbenv-bundler-ruby-version          rubyfmt

ruby                                homebrew/portable-ruby/portable-ruby

ruby-build
```

You can then install an appropriate version:

```
> brew install ruby@3.0
```

Linux

The directions for Linux are the same of the instructions for rbenv in the WSL section above.

Install Rails

Now that you have a working version of Ruby, you're set to install Rails.

Windows and Linux

```
> gem install rails

Fetching tzinfo-2.0.5.gem

Fetching zeitwerk-2.6.0.gem

[…]

Done installing documentation for zeitwerk, thor, method_source, concurrent-
ruby, tzinfo, i18n, activesupport, nokogiri, crass, loofah, rails-html-
sanitizer, rails-dom-testing, rack, rack-test, erubi, builder, actionview,
actionpack, railties, mini_mime, marcel, activemodel, activerecord,
globalid, activejob, activestorage, actiontext, mail, actionmailer,
actionmailbox, websocket-extensions, websocket-driver, nio4r, actioncable,
rails after 66 seconds

35 gems installed

> which rails

/Users/stevebeaty/.rbenv/shims/rails

> rails --version

Rails 7.0.3.1
```

And yes, you might have used a software manager (apt) to install a software manager (rbenv) to install a software manager (gem) to install Rails.

OSX

Make sure you're using the version you install via Homebrew; if you're still pointing to /usr/bin/ruby, you won't be able to install rails and will get a permissions error:

```
> ruby --version

ruby 2.6.3p62 (2019-04-16 revision 67580) [universal.x86_64-darwin20]

> gem install rails

Fetching crass-1.0.6.gem

Fetching rack-2.2.4.gem
```

```
[…]

ERROR:  While executing gem ... (Gem::FilePermissionError)

  You don't have write permissions for the /Library/Ruby/Gems/2.6.0
directory.
```

OSX may suggest you do

```
> sudo gem install rails
```

Do not do this! You can break the system administration tools, something that can be difficult to recover from. Instead, make sure your PATH variable is set to have /usr/local or /opt/homebrew before /usr/bin and then the install of Rails should proceed without the need to sudo.

Test

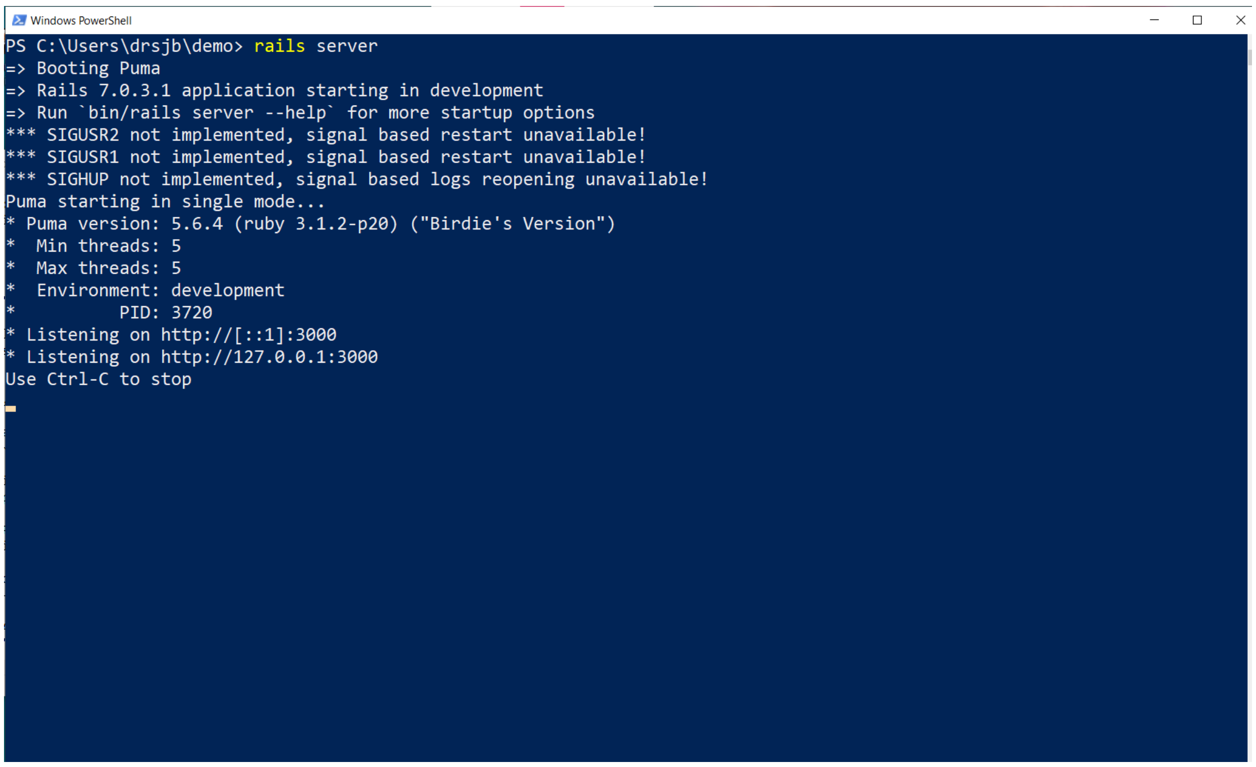
You can now create a very simple application.

> rails new demo

> cd demo

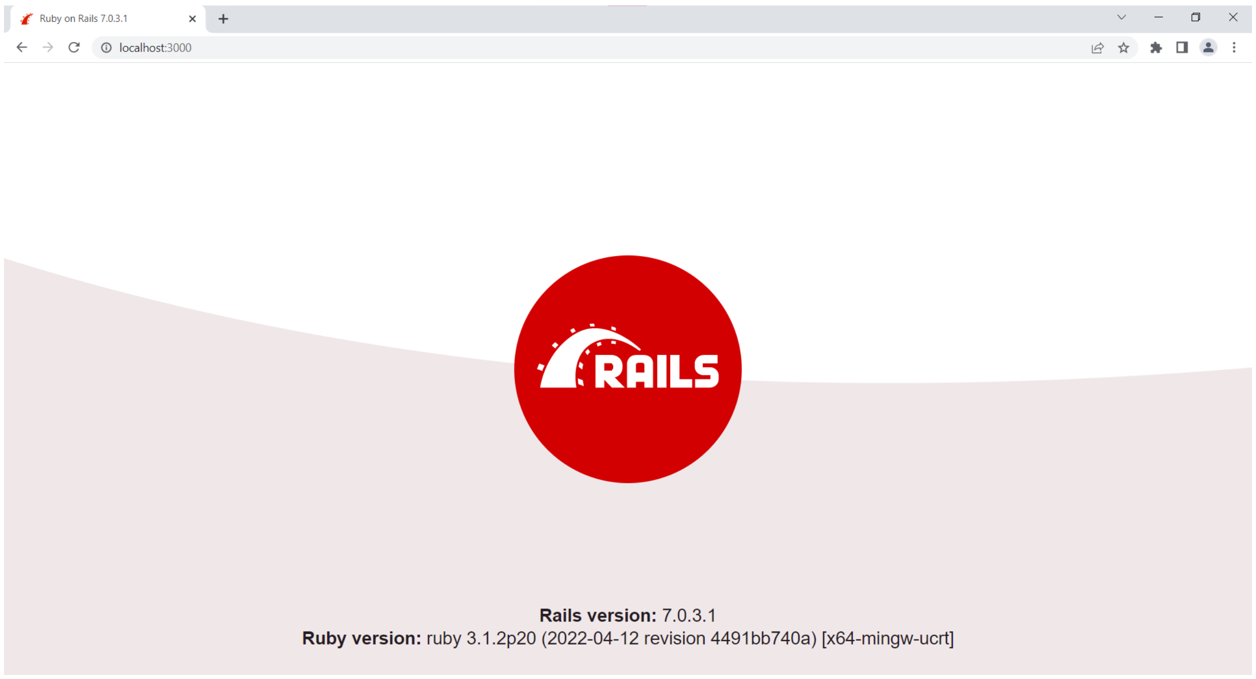
> rails server

Here's what it should look like in PowerShell, though it will look similar on all platforms.



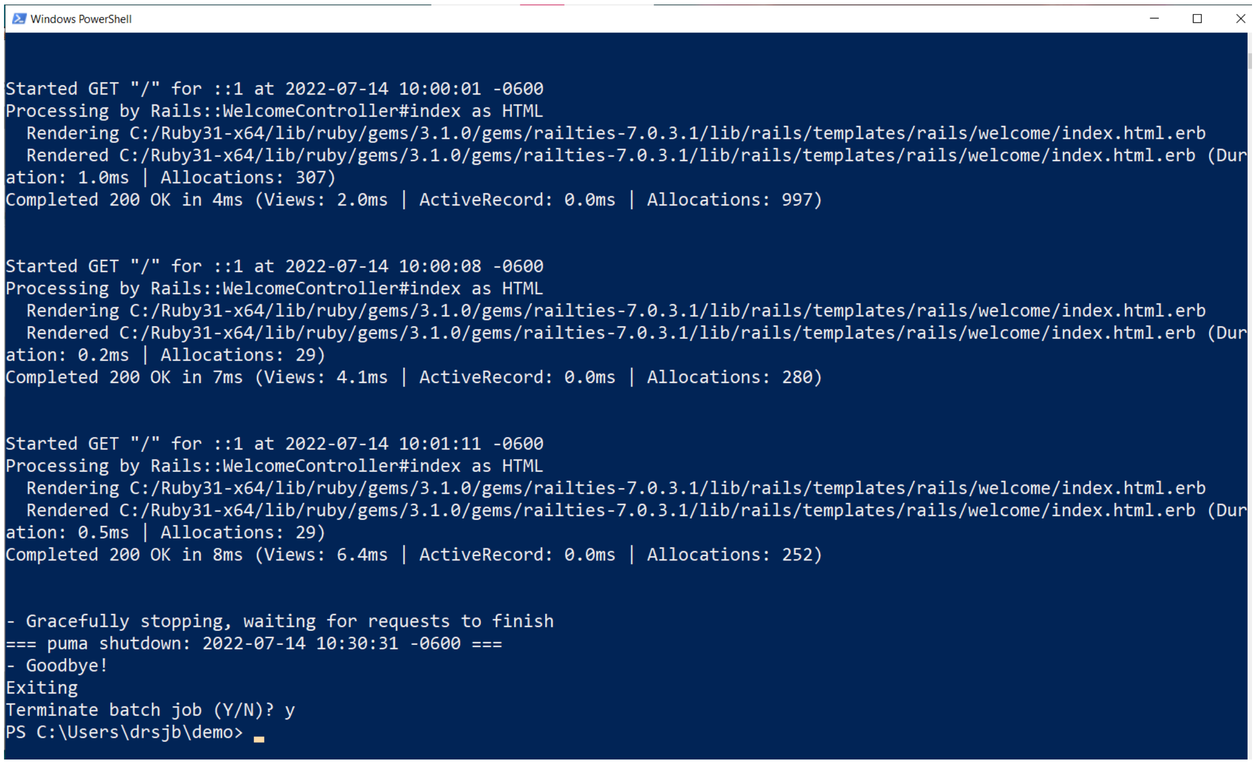
```
PS C:\Users\drsjb\demo> rails server
=> Booting Puma
=> Rails 7.0.3.1 application starting in development
=> Run `bin/rails server --help` for more startup options
*** SIGUSR2 not implemented, signal based restart unavailable!
*** SIGUSR1 not implemented, signal based restart unavailable!
*** SIGHUP not implemented, signal based logs reopening unavailable!
Puma starting in single mode...
* Puma version: 5.6.4 (ruby 3.1.2-p20) ("Birdie's Version")
*  Min threads: 5
*  Max threads: 5
*  Environment: development
*          PID: 3720
* Listening on http://[::1]:3000
* Listening on http://127.0.0.1:3000
Use Ctrl-C to stop
```

Point your browser at http://127.0.0.1:3000 and that should show the following.



Rails version: 7.0.3.1
Ruby version: ruby 3.1.2p20 (2022-04-12 revision 4491bb740a) [x64-mingw-ucrt]

Once you're doing experimenting, type Control-C in the terminal to stop the server; here's what that should look like.



```
Started GET "/" for ::1 at 2022-07-14 10:00:01 -0600
Processing by Rails::WelcomeController#index as HTML
  Rendering C:/Ruby31-x64/lib/ruby/gems/3.1.0/gems/railties-7.0.3.1/lib/rails/templates/rails/welcome/index.html.erb
  Rendered C:/Ruby31-x64/lib/ruby/gems/3.1.0/gems/railties-7.0.3.1/lib/rails/templates/rails/welcome/index.html.erb (Dur
ation: 1.0ms | Allocations: 307)
Completed 200 OK in 4ms (Views: 2.0ms | ActiveRecord: 0.0ms | Allocations: 997)


Started GET "/" for ::1 at 2022-07-14 10:00:08 -0600
Processing by Rails::WelcomeController#index as HTML
  Rendering C:/Ruby31-x64/lib/ruby/gems/3.1.0/gems/railties-7.0.3.1/lib/rails/templates/rails/welcome/index.html.erb
  Rendered C:/Ruby31-x64/lib/ruby/gems/3.1.0/gems/railties-7.0.3.1/lib/rails/templates/rails/welcome/index.html.erb (Dur
ation: 0.2ms | Allocations: 29)
Completed 200 OK in 7ms (Views: 4.1ms | ActiveRecord: 0.0ms | Allocations: 280)


Started GET "/" for ::1 at 2022-07-14 10:01:11 -0600
Processing by Rails::WelcomeController#index as HTML
  Rendering C:/Ruby31-x64/lib/ruby/gems/3.1.0/gems/railties-7.0.3.1/lib/rails/templates/rails/welcome/index.html.erb
  Rendered C:/Ruby31-x64/lib/ruby/gems/3.1.0/gems/railties-7.0.3.1/lib/rails/templates/rails/welcome/index.html.erb (Dur
ation: 0.5ms | Allocations: 29)
Completed 200 OK in 8ms (Views: 6.4ms | ActiveRecord: 0.0ms | Allocations: 252)


- Gracefully stopping, waiting for requests to finish
=== puma shutdown: 2022-07-14 10:30:31 -0600 ===
- Goodbye!
Exiting
Terminate batch job (Y/N)? y
PS C:\Users\drsjb\demo>
```

Cross-Platform

Web-Based

There are sites that allow you to develop RoR applications via a web browser that are either free or cost very little. Here are several, though it must be noted that these types of sites come and go through the years.

- https://replit.com/
- https://codeanywhere.com/

Docker

Docker (https://www.docker.com/) is a widely used container-based approach to development and deployment. It allows the creation of lightweight “VMs” that can run easily on many different platforms. If you already have it installed or are interested in exploring its capabilities, this approach is for you. There is a simple Docker image for developing RoR applications in the central Docker repository. To run:

```
docker run -it -p 3000:3000 -v ${PWD}:/home drsjb80/rails-dev
```

You'll have to run the Rails commands in the container and in the /home directory, but then you can point your editor/IDE at the created directory on the host machine. It's easiest to run the Rails server bound to all IP addresses via

```
rails server -b 0.0.0.0
```

For example, in the container:

```
# cd home
/home # rails new demo
      create
      create  README.md
      create  Rakefile
      create  .ruby-version
[...]
Appending: pin "@hotwired/stimulus-loading", to: "stimulus-loading.js",
preload: true
      append  config/importmap.rb
Pin all controllers
Appending: pin_all_from "app/javascript/controllers", under: "controllers"
      append  config/importmap.rb
/home # cd demo/
/home/demo # rails server -b 0.0.0.0
=> Booting Puma
=> Rails 7.0.3.1 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.6.4 (ruby 3.1.2-p20) ("Birdie's Version")
*  Min threads: 5
*  Max threads: 5
*  Environment: development
*          PID: 403
* Listening on http://0.0.0.0:3000
Use Ctrl-C to stop
```

There will now be a "demo" folder in the host machine where you ran the “docker run” command with the Rails application that you can modify. You can point your browser at http://127.0.0.1:3000 to see the running Rails server.