

Ace on Tech Blog Article

How to: Set up your Mac to develop Ruby on Rails

Web development of is an area of general heterogeneousness when it comes to the tools used to develop for it. 'Regular' standalone software development is often characterized by its standardized development regime, with its compilers and IDEs (Integrated Development Environments) defining the stage.



Enter the world of web development and matters are startlingly different. For most scripting languages, be it client-side or server-side, there are no de facto IDEs, nor are there compilers to deal with (generally a nice side-effect). While this gives you a lot of freedom, it may also be hair-raising and confusing, especially to beginners.

Since my interest for Ruby on Rails has recently been refueled I set out to deepen my knowledge on the matter. Soon enough, I was faced with setting up my computer to accommodate Ruby on Rails development. Having made OS X my primary (though not sole) operating system, I was delighted to see Mac OS X had nearly all the tools pre-installed and could be updated effortlessly.

Multi-platform

Ruby on Rails - or RoR as it's commonly abbreviated - is a platform that can be run and developed for on practically any contemporary operating system. This includes Mac OS X, Windows and most any Linux-derivative.

I've set up RoR on all three and will be sharing my experiences (and hurdles I've come across) with you, since things may not always work out at first glance depending on which OS you use.

RoR Recommended Defaults

Ruby on Rails is a fairly new platform and therefore has undergone numerous significant changes throughout its short lifetime. These changes influence how you need to set up your computer.

With this in mind, it may be handy to list up the current versions of the Ruby on Rails' components (as of February 2009):

- Ruby: 1.8.6 as installed by OS X by default (latest stable release: 1.9.1)
- Rails: 2.2.2
- RubyGem: 1.3.1
- SQLite: 3.4.0

SQLite is now the default development database because it is light-weight and requires no configuration whatsoever. While other database solutions like MySQL and ORACLE aren't, SQLite is completely portable, because it's essentially a flat-file embedded in the Rails application directory.

Set up RoR on your Mac

Now, without further ado, let's set up our Mac for RoR development.

Using OS X 10.5, you will have most of the prerequisites installed, but they will most likely require some updating to bring them up to speed with the rest of the world. Rails uses its own terminal-based package manager called RubyGem. RubyGem is pre-installed in a version just functional enough to update itself and other RoR-components.

1. So, before you do anything else, update RubyGem like so (don't type the '\$', it's a marker preceding every terminal command-line):
\$ sudo gem update --system
You may skip to step 3 if you're okay with the pre-installed version of Ruby.
2. If you're running an up-to-date version of OS X (10.5.0 and up), there's **no real need to update your Ruby installation**. Moreover, there's no standardized way to do so on the Mac, so if you really want to do this, you will have to resort to [MacPorts](#). After its installation, type this in a terminal:
\$ sudo port install ruby
3. Now, install Rails using *gem*:
\$ sudo gem install rails
4. While you're at it, update *rake*, Rails' equivalent of C's *make*:
\$ sudo gem update rake
5. In a final effort, why not update SQLite as well:
\$ sudo update sqlite3-ruby

Et voilà, you're ready to go.

That as the easy part, now get to developing!

Windows-users will get their shot of Rails goodness in the near-future. Stay tuned.