

# Ace on Tech Blog Article

## How to: Set up your PC 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 (which is generally a pleasant 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. Unlike OS X, Microsoft Windows doesn't come with Ruby or Rails baked right in, so we'll need look at the options for installing it locally. As it turns out, things can turn out to become slightly complicated if you're on Windows.*

### 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: latest 1.8 series stable version by default (latest 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 lightweight 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 located in the Rails application directory.

### Set up RoR on your PC

I'll be assuming your computer is reasonably up to date with Windows Updates and the like. My testing machine was running Windows Vista SP1 and had no RoR-related components installed.

Realistically, there are two ways to get Ruby on Rails on your PC: a full native install of Ruby, Rails, and SQLite3 or the prepackaged (and recommended) InstantRails. Either approach has its own pros and cons, so you'll have to decide for yourself which road you wish.

I prefer the prepackaged option merely because I was utterly incapable of getting SQLite3 recognized by Rails. I've searched high and low, but to no avail. I will describe the process, though, if you oh so dearly wish to venture into it, but as I said, if you're unfamiliar (or intentionally ignorant) on the matter, I recommend going with InstantRails.

The pros of InstantRails:

- Instant gratification
- Zero config. Zero Hassle.

The cons:

- Not up-to-date out-of-the-box, will need some updating after installation.
- Comes with Apache Webserver and MySQL unnecessarily.
- Mandatory use of InstantRails command prompt.

The Rails development team recommends using Webrick/Mongrel as the development webserver and SQLite3 as the database server, because of their sheer ease of use. If you decide to follow their guidelines (which you should as a starting RoR-developer), you'll end up never using Apache and MySQL. Whilst this is not a problem in itself, one might argue that they're dispensable cruft.

I say, don't give a hoot and get InstantRails running a few concise steps:

1. Grab the latest zip-file from SourceForge: [http://rubyforge.org/frs/?group\\_id=904](http://rubyforge.org/frs/?group_id=904).
2. Decompress it into a directory on your disk, e.g.:  
*C:\InstantRails*
3. Start InstantRails:  
*open InstantRails.exe from the install folder*
4. Open the InstantRails command-line from the InstantRails command-window and type:  
*gem update --system*  
*gem update ruby*  
*gem update rails*  
*gem update sqlite3*

After some command-line churning invoked by the commands above, your Rails installation should be ready to go. As mentioned, Ruby on Rails commands will only work from within the InstantRails command-prompt. It is also recommended you place all of your Rails Apps in the designated folder (e.g. C:\InstantRail\rails\_apps).

If all goes well, you should be ready to go.

### **Full-on native install**

If you prefer to do a component-by-component installation of Ruby on Rails, you can. At first glance, this may be the preferred method since you get the opportunity to leave out unnecessary stuff (like Apache). I myself, have encountered insurmountable issues with SQLite3 on Windows Vista, preventing me from completing the process.

1. Get the Ruby installer from [www.ruby-lang.org](http://www.ruby-lang.org)
2. Once complete, update gem:  
*gem update --system*
3. Update Ruby and Rails:  
*gem update ruby*  
*gem update rails*
4. Go to <http://www.sqlite.org/download.html>, download the latest Windows ZIP, and unpack it into the 'ruby/bin' directory (e.g. C:\InstantRails\ruby\bin)
5. Add the 3 files you've just unpacked (.exe, .dll, .def) to your system's path, [here's how](#).

6. Install the SQLite3 driver to Rails:

```
gem install sqlite3-ruby
```

If you come across no hiccups, you should have systemwide access to Rails.

### **Conclusion**

In retrospect, it's fairly easy to get Ruby on Rails running on Windows, save for the fact it's a little convoluted. Once more, if you want fast access to RoR, I'd recommend you go with the InstantRails-approach. It is the most automatic install-procedure around, and Windows-users like automatic, and so do I.