

Hivelogic

[Home](#) [Archives](#) [Tags](#) [Links](#) [About](#) [Projects](#) [Contact](#) [Feeds](#)



Building Ruby, Rails, LightTPD, and MySQL on Tiger

[Technology](#) | [Web Design](#) | [Development](#) | [Mac OS X](#) | [Ruby on Rails](#) | 12.01.05 - 09:22 PM

Ads by Yahoo!

[Gold Ruby Earring](#)

Shop for great deals on and...
www.monstermarketplace.com

[Hotels.com - Save on Hotel...](#)

Reserve your room in...
hotels.com

What follows are updated instructions for manually building and installing [Ruby](#), [Ruby on Rails](#), [MySQL](#), and [LightTPD](#) on Mac OS X 10.4 (Tiger).

Compiling and installing these tools this way is well worth the effort, as the end result delivers an easy-to-upgrade, system-independent, stand-alone development platform that is impervious to potential problems that can be caused by system updates, operating system upgrades, etc. These issues and additional background information about *why* one might roll their own tools in this fashion is detailed in the article, [Using /usr/local/](#), which could be considered a prerequisite for this task.

The Concept

Basically, what we're going to do here is download a bunch of open-source tools (some of which rely upon each other to work), configure them, compile them, and install them, one by one, until we have everything we need for a Mac OS X machine to run pretty much any Ruby on Rails application.

What's Needed

A few things are needed to get this going:

1. Mac OS X 10.4 (Tiger)
2. Xcode 2.0 or newer (ideally 2.1)
3. Willingness to type commands into the Terminal application exactly as they appear here
4. A tasty beverage to enjoy while things compile

A Quick Warning

While it's unlikely anything we do here might do any kind of damage to the system, it's good advice to

have a current backup of everything, just in case. The Narrator doesn't take any responsibility for anything that results from following these instructions. We're following these instructions at our own risk.

Setting Up

Open the Terminal application. It can be found in the `/Applications/Utilities` folder.

Each of the lines below appearing in monospaced type should be entered into Terminal, and be followed by the Return key. But everybody knew that already.

Create a folder to contain all of the downloaded files and their respective build folders. For these examples, we'll create a folder called `src` in the root of our home folder, and change directories into that folder. It will be our workspace:

```
mkdir src
cd src
```

It doesn't really matter where this folder actually lives. We've created it in the home folder, but it could be on the Desktop, or in `/usr/local/src` for example. All operations should take place there.

You'll download and compile everything from right here.

Paths

We need to make sure that our path is set to look for files in `/usr/local` (also the place where we'll be installing the tools) before looking anywhere else.

To see if the path has been set properly, we can check the contents of the `.bash_login` file—a special, hidden file in the root of our home folder—for a `PATH` line using a text editor. [TextMate](#), [TextWrangler](#), [BBEdit](#), and [vi](#) are all perfectly good options. To open the file with TextMate, for example, we can type:

```
mate ~/.bash_login
```

This will open the file if it already exists, or open a blank file if it doesn't. In either case, add the following line at the very end of the file:

```
export PATH="/usr/local/bin:/usr/local/sbin:$PATH"
```

Now save and close the file.

To make sure the changes are picked up correctly, we now need to execute the file with the following command:

```
~/.bash_login
```

It's likely there will be no response from the shell here, just the prompt, but that's OK, the changes have been picked up and we're ready to move on.

Ruby

We're ready to start the process. Just type (or cut and paste) each one of the following lines into

Terminal, one by one. When one line finishes (some will take a while and dump a lot of information to the screen), enter the next one.

The first time we run the `sudo` command, and possible again later, we may be prompted for a password. We'll just enter our own password here, and the process will continue.

We'll start by installing Readline, a prerequisite for Ruby on OS X systems:

```
curl -O ftp://ftp.gnu.org/gnu/readline/readline-5.0.tar.gz
tar xzvf readline-5.0.tar.gz
cd readline-5.0
./configure --prefix=/usr/local
sed -e 's/--dynamic/--dynamiclib/' shlib/Makefile > shlib/Makefile.new
mv shlib/Makefile.new shlib/Makefile
make
sudo make install
cd ..
```

Next up, we'll download and install Ruby itself:

```
curl -O ftp://ftp.ruby-lang.org/pub/ruby/ruby-1.8.2.tar.gz
tar xzvf ruby-1.8.2.tar.gz
cd ruby-1.8.2
./configure --prefix=/usr/local --with-readline-dir=/usr/local
make
sudo make install
cd ..
```

RubyGems

RubyGems is a handy command-line tool for managing the installation of Ruby packages, like Rails.

```
curl -O http://rubyforge.org/frs/download.php/3700/rubygems-0.8.10.tgz
tar xzvf rubygems-0.8.10.tgz
cd rubygems-0.8.10
sudo /usr/local/bin/ruby setup.rb
cd ..
```

Ruby on Rails

With RubyGems installed, Rails is a simple, one-line install:

```
sudo gem install rails --include-dependencies
```

It's possible that we may see an "RDoc failure" error here. The error is actually nothing to worry about. Just re-run the command above, or don't. Things should be fine either way. Really.

FastCGI

FastCGI is an extension to CGI (Ruby on Rails is essentially a collection of CGI scripts) that provides high performance without the limitations of server specific APIs. Don't worry if that doesn't make perfect sense. Just so long as it's installed on the system:

```
curl -O http://www.fastcgi.com/dist/fcgi-2.4.0.tar.gz
tar xzvf fcgi-2.4.0.tar.gz
```

```
cd fcgi-2.4.0
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

We'll also need to add the Ruby-FastCGI bindings

```
curl -O http://sugi.nemui.org/pub/ruby/fcgi/ruby-fcgi-0.8.6.tar.gz
tar xzvf ruby-fcgi-0.8.6.tar.gz
cd ruby-fcgi-0.8.6
/usr/local/bin/ruby install.rb config --prefix=/usr/local
/usr/local/bin/ruby install.rb setup
sudo /usr/local/bin/ruby install.rb install
cd ..
```

We're almost done with FastCGI. We just need the Ruby-FCGI Gem—another nice one-line installation:

```
sudo gem install fcgi
```

LightTPD

LightTPD is an open-source webserver designed with security, speed, compliance, and flexibility in mind. It's great for serving up Rails applications, and Rails 0.14 and newer is already setup to use it if it's on the system.

But before we can compile it, we need to build its prerequisite, the PCRE library, a set of functions that implement regular expression pattern matching using the same syntax and semantics as Perl 5:

```
curl -O ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-6.4.tar.gz
tar xzvf pcre-6.4.tar.gz
cd pcre-6.4
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

With PCRE installed, we're ready to move on to LightTPD:

```
curl -O http://lighttpd.net/download/lighttpd-1.4.8.tar.gz
tar xzvf lighttpd-1.4.8.tar.gz
cd lighttpd-1.4.8
./configure --prefix=/usr/local --with-pcre=/usr/local
make
sudo make install
cd ..
```

MySQL

While it's possible to compile and install MySQL ourselves, using the OS X MySQL package is actually advantageous. Not only is the install much faster and easier, but the package includes a handy startup item and a preference panel, and the binary is tuned by the MySQL team for OS X.

Even better, the package installs MySQL right into the `/usr/local/` folder, just like it should.

The install still requires a few steps:

1. Download the [MySQL 4.1 package for OS X](#)
2. Double-click the drive image to mount it
3. Locate the MySQL installer (a file named `mysql-standard-4.1.15-apple-darwin8.2.0-powerpc.pkg`) and run it, authenticating as needed
4. Double-click `MySQLStartupItem.pkg`, authenticate, and let it install
5. Double-click `MySQL.prefPane` and install it, deciding whether to make it available to just the current user, or for all system users

Once the install is complete, start the MySQL server using the newly-installed control panel.

Note: MySQL installs with a default user of `root` which has no password. Please read this page about [MySQL usernames and passwords](#) and set a good one.

MySQL Native Bindings (Optional)

This step is an optional one, but the [performance gains](#) seem to make it worth the extra step. Thanks to RubyGems, it's another one-line install:

```
sudo gem install mysql -- --with-mysql-dir=/usr/local/mysql
```

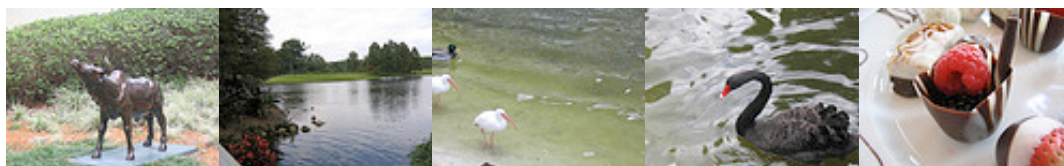
We're Done

Phew! It's over. We now have a properly located installation of Ruby, Ruby on Rails, MySQL, and LightTPD.

So ... what's the hold-up? Start writing those Web 2.0 applications already!

[More recent articles »](#)

Photostream via [flickr.com](#)



Links & News

[Dr. Moreau's Pet Mouse](#)

12.13.05 - 01:14 PM | [Permalink](#)

Only the most foolish of mice *wouldn't* want human embryonic stem cells injected into their brains.

[Polo 10-Pack](#)

12.12.05 - 05:31 PM | [Permalink](#)

Polo is definitive (and required) Hivelogic attire. And now, at long last, 10-packs of the classic-fit shirts are available.

[A Ruby/Rails/Lighttpd/MySQL Tiger Script](#)

12.11.05 - 06:55 PM | [Permalink](#)

My friend Geoffrey has converted the [Building Ruby, Rails, LightTPD, and MySQL on Tiger article](#) into a handy shell script.

[More links »](#)

Copyright © 2005 Dan Benjamin. [Contact The Narrator](#). Hosted by [TextDrive](#). Don't steal this content.



[» Blogs that link here](#)

