Installing Rails 2.3 Under Windows XP and Apache 2.2

Scott Taylor
Tailor Made Software
August 9, 2011
Version 1.0

1.0 Introduction

Ruby On Rails (aka just "Rails") is a modern scripting system that allows easy development of "Web 2.0" type applications. It relies heavily on Dynamic HTML (dHTML) and AJAX, a form of JavaScript that provides asynchronous communication between server and client.

The following instructions will install Rails 2.3.x under Apache 2.2 on Windows XP or higher (32-bit or 64-bit). They should work for any version of Windows newer than XP, but were tested using Windows XP 64.

Windows applications are downloaded and then installed using MSI or other Windows Installer files. Ruby Gems will be installed from pre-downloaded versions.

Instructions are given for file serving using Apache and load balancing using Mongrel Clusters. Unlike linux, Passenger, the other major method of load balancing, is not available on Windows. Microsoft IIS can also be used in place of Apache.

2. Prerequisites

There are several packages that are required to run and install Rails, including the Ruby Interpreter (the system that processes the Ruby language files), RubyGems (the installation system for Ruby modules) and Apache 2.2 (the web server).

2.1 Apache

First we will install Apache 2 or verify that Apache 2 is installed correctly.

Apache comes in two versions: with and without OpenSSL. You can download the two versions from http://httpd.apache.org/download.cgi#apache22

```
With OpenSSL:    httpd-2.2.19-win32-x86-openssl-0.9.8r.msi
Without OpenSSL: httpd-2.2.19-win32-x86-no_ssl.msi
```

Note: for this and all instances in these instructions where a download is given, the version number may be newer.

Download Apache and double click on the msi package to install it. Unless there is a good reason, use the default locations.

2.2 Ruby

Under Windows there is only one options for installing Ruby: "standard Ruby". Standard Ruby is the normal Ruby interpreter. Enterprise Ruby is a free version that is developed by Phusion, a Dutch software company. It is faster and manages memory better than standard Ruby, but it is not available on Windows, only Linux.

2.2.1 Install

Download ruby from http://rubyforge.org/projects/rubyinstaller - choose "RubyInstaller" not "One Click Installer". Follow the directions to install it. It will give you options to update the path and associate .rb files with Ruby. It is suggested that you do both.

Note: *You should install Ruby 1.8.6 or 1.8.7 (preferred) not Ruby 1.9*. Ruby 1.9 has some problems with Rails 2.3.

After installation is complete, run "ruby –v" from the command line to make sure the installation worked correctly.

2.3 RubyGems

RubyGems is an installation system, sort of like yum or apt, that is used by Ruby modules. RubyGems is installed by RubyInstaller, so you do not have to do anything extra. You can run "gem update –system" to make sure you have the latest version.

2.4 Uninstall Rails 3.0

A clean Windows will not have Rails installed, but if you have an older system you may have an incorrect version of Rails installed. We are using Rails 2.3.11.

You will use gem to uninstall the existing version of Rails and dependant packages. The command is:

```
gem uninstall abc –v=x.y.z
```

where "abc" is the name of the package to be removed and "x.y.z" is the version. If there is more than one version of the package installed and you do not list the version you want uninstalled, it will prompt you for which version to remove, and give you a choice to remove all versions.

The command "gem list" will give you a list of all gems installed. A sample after rails 2.3.11 has been installed:

```
actionmailer (3.0.9, 2.3.11)
actionpack (3.0.9, 2.3.11)
activemodel (3.0.9)
activerecord (3.0.9, 2.3.11)
activeresource (3.0.9, 2.3.11)
activesupport (3.0.9, 2.3.11)
arel (2.0.10)
builder (2.1.2)
bundler (1.0.15)
daemon_controller (0.2.6)
erubis (2.6.6)
fastthread (1.0.7)
formtastic (1.2.4)
i18n (0.6.0, 0.5.0)
mail (2.2.19)
mime-types (1.16)
mysql (2.8.1)
passenger (3.0.7)
polyglot (0.3.1)
rack (1.3.0, 1.2.3, 1.1.2)
rack-mount (0.6.14)
rack-test (0.5.7)
rails (3.0.9, 2.3.11)
railties (3.0.9)
rake (0.9.2)
```

Each item lists the name followed by the version number(s). For instance, activerecord has versions 3.0.9 and 2.3.11 installed.

You need to remove the following if present:

```
actionmailer 3.0.9
actionpack 3.0.9
activemodel 3.0.9
activerecord 3.0.9
activeresource 3.0.9
activesupport 3.0.9
builder 2.1.2
bundler 1.0.15
passenger 3.0.7
rack 1.3.0 and 1.2.3
rack-mount 0.6.14
rack-test 0.5.7
rails 3.0.9
railties 3.0.9
```

3. Install Rails

3.1 Installation

Installation of Rails is straightforward. The easiest method is to copy the gem files to a directory and run RubyGems from that directory. You will need to install the following gem files:

- activerecord-2.3.11.gem
- activeresource-2.3.11.gem
- actionmailer-2.3.11.gem
- activesupport-2.3.11.gem
- actionpack-2.3.11.gem
- rack-1.1.0.gem
- rake-0.8.7.gem

These are installed in two steps. First run:

```
gem install rack-1.1.0.gem
gem install rake-0.8.7.gem
```

then run:

```
gem install rails -v=2.3.11
```

The rails install will install all of the "active" and "action" gems.

You should then install the other gems needs by Visual Query:

```
gem install i18n-0.6.0.gem
gem install formtastic-1.2.4.gem
```

Note: *when you install formtastic on a Rails 2.3 system it will complain that activesupport 3.0 or higher is required. That is a problem with the gem installer and is not really true. Activesupport is part of Rails 3 and is not required for Rails 2. If you run "gem list" you will see that formtastic 1.2.4 was actually installed despite the "error" notice.*

You will need to install the database connector unless you are using SqlLite 3. For MySQL install use:

```
gem install mysql
```

If you are using MySQL 5.1 you will need to install the MySQL 5.0 client library instead. Thankfully you do not have to replace MySQL 5.1 as your database management system, you just have to put the MySQL 5.0 client library in your \Ruby\bin directory. You can download the MySQL 5.0 Client Library from

```
http://instantrails.rubyforge.org/svn/trunk/InstantRails-
win/InstantRails/mysql/bin/libmySQL.dll
```

If you try using the MySQL 5.1 DLL you will get an "Argument Error in AbcController#index" (where "Abc" is the name of the function you are trying to run; i.e. Employees, Rooms, etc) when you try running Visual Query.

If you are using Oracle you will need to install the Oracle Enhanced Driver:

```
gem install ruby-oci8
gem install activerecord-oracle_enhanced-adapter-1.3.2.gem
```

3.2 Testing Rails

With Rails and Ruby installed it is a good idea to test the installation so far. That is easiest by creating a simple rails application and running it. The instructions below use mysql for the database and call the project "rtest". You should modify them as needed.

Create a directory for the rails project and then run:

```
rails –d mysql rtest
```

Create a database in the Mysql database:

```
mysql –u root
mysql> create database rtest;
```

Modify the config/database.yml file to point to the new database:

```
development:
        adapter: mysql
        encoding: utf8
        reconnect: false
        database: rtest
        pool: 5
        username: root
        password:  jpmc
        host: backset
```

Create the database table using rails:

```
script/generate scaffold post name:string
```

This will create the table "post" in the "rtest" database.

If then you start the rails server using "ruby script/server" and go to the main page "localhost:3000" it will display the Rails equivalent of "Hello world!". Click on the "About your applications environment" link and it should display the information about your setup of rails. It should be something like this:

```
Ruby version      1.8.7 (x86_64-linux)
RubyGems version  1.5.0
Rack version      1.1
Rails version     2.3.11
Active Record version   2.3.11
Active Resource version 2.3.11
Action Mailer version   2.3.11
Active Support version  2.3.11
Application root  /www/rtest
Environment development
Database adapter  mysql
Database schema version 0
```

3.3 Required Gems

Visual Query requires several gems to run correctly. Thankfully these gems will be installed in one step using the following command:

```
rake gems:install
```

To see a list of required gems prior to installation run:

```
rake gems
```

They can be installed manually using:

```
prawn             gem install prawn
```

3.4 Required Plugins

Visual Query requires several plugins to run correctly. These plugins will be delivered already installed in the public/vendor/plugins directory. If they need to be reinstalled for any reason the current plugins can be found at:

```
auto_complete     https://github.com/rails/auto_complete.git
prawn_to          http://github.com/thorny-sun/prawnto.git
paperclip         https://github.com/thoughtbot/paperclip.git
```

4. Load Balancing Software

In order to run more than one instance of the Rails application we will need to install multiple server instances and load balancing software. Under Windows the only options are Mongrel and FastCGI. Mongrel is the more modern and definitely recommended solution. On Linux we recommend Phusion Passenger, but that system does not run on Windows.

4.1 Mongrel

Mongrel, and its associated process Mongrel Cluster, work in conjunction with Apache to provide load balancing and multiple process support. Mongrel is actually a web server written in Ruby that is optimized for running Ruby-based processes like Rails.

Mongrel Cluster allows multiple instances of Mongrel to run in parallel. Apache is used to provide load balancing using *mod_proxy_balancer* and to serve static pages, while Mongrel is used to serve Ruby-based dynamic pages.

4.1.1 Installation

Mongrel and Mongrel Cluster are installed from their gems. Use the commands:

```
gem install mongrel –v=1.1.5
gem install mongrel_cluster –v=1.0.5
```

4.1.2 Configuration

Mongrel has no specific configuration requirements. Everything is configured using Mongrel Cluster and Mongrel Service.

4.1.3 Mongrel Cluster

To configure Mongrel Cluster you need to know the port to use (we will using 3000), the number of instances (we will start with 3) and which environment is being run (we will use development for now).

First change directory to the home directory for the application and then run mongreal cluster configure:

```
cd /www/public_html/testapp
mongrel_rails cluster::configure –e development –p 3000 –N 3 –c
/www/public_html/testapp –a 192.168.1.104
```

Of course you should use the appropriate values for the www directory and the URL.

Note that this will create the file `config/mogrel_cluster.yml`. Make sure you have write permission to the config directory.

You can manually start, restart and stop the cluster (from the document root directory) using:

```
mongrel_rails cluster::start
mongrel_rails cluster::restart
mongrel_rails cluster::stop
```

4.1.4 Mongrel Service

Obviously having to start Mongrel by hand each time the server is rebooted is not practical. Mongrel Service allows Mongrel to be started automatically as a Windows Service.

Install Mongrel Service using the appropriate Windows version for the gem:

```
gem install mongrel_service --platform x86-mswin32
gem install mongrel_service --platform x86-mswin64
```

Additionally, to actually install the service you need to run this command in your Rails application directory:

```
mongrel_rails service::install --name MyApp -e production -p 3001 -a 0.0.0.0
```

4.2 Windows Host File

You will need to add the virtual server to the Windows Host file. By default this is in C:\WINDOWS\system32\drivers\etc\hosts. The format is very simple – the IP address for the server followed by the virtual host name.

```
192.168.1.101           jpmcvq.com
```

4.3 Apache Configuration

4.3.1 HTTPD.CONF

The main Apache configuration file needs to be modified to ensure the proxy modules are enabled. Ensure that the following are uncommented in the httpd.conf file:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule vhost_alias_module modules/mod_vhost_alias.so
```

4.3.2 HTTPD-VHOSTS.CONF

The final step is to configure apache to use mongrel. This can be done in either the c:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf\httpd.conf or the c:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf\extra\httpd-vhosts.conf file depending on how you have configured Apache (note the path name may vary depending on where Apache is installed).

```
<VirtualHost *:80>
    ServerAdmin webmaster@www.jpmcvq.com
    ServerName jpmcvq.com
    ServerAlias www.jpmcvq.com

    DocumentRoot “c:/www/rtest/public”

    RewriteEngine On
    RewriteCond %{DOCUMENT_ROOT}/%{REQUEST_FILENAME} !-f
    RewriteRule ^/(.*)$ balancer://mongrel1%{REQUEST_URI} [P,QSA,L]

    <proxy balancer://mongrel1>
        BalancerMember http://192.168.1.104:3000
```

```
        BalancerMember http://192.168.1.104:3001
        BalancerMember http://192.168.1.104:3002
    </proxy>


    ProxyPass / balancer://mongrel1/
    ProxyPassReverse / balancer://mongrel1/
    ProxyPreserveHost on

    <Proxy *>
        Order deny,allow
        Allow from all
    </Proxy>

    ErrorLog /www/rtest/log/error.log
    CustomLog /www/rtest/log/access.log combined
</VirtualHost>
```