



Ruby on Rails

Matt Dees



All trademarks used herein are the sole property of their respective owners.



Introduction

- How Ruby on Rails Works
- cPanel's interaction with Ruby on Rails
- Administrating Ruby on Rails
- Troubleshooting Ruby on Rails





What is Ruby on Rails?

A Web Application Framework aimed towards the rapid development and deployment of Dynamic Web 2.0 Applications





Ruby

A Programmer's Best Friend

- »» Interpreted Programming Language
- »» Web Applications are done through either Rails or as a straight CGI application
- »» Every part of the Ruby on Rails system is dependent on ruby working correctly



Gems are Ruby modules

- Either compiled or interpreted Ruby code
- Gems can be full applications or libraries for Ruby programs
- Managed by the “gem” command

Rails is a framework for creating Ruby applications and provides several different pieces of functionality

- »» Rails exists for multiple programming languages
- »» Is a gem
- »» Consists of several gems used for handling different functions
- »» Different versions of this exist, each application requires a specific version



Rails Continued



- »» Action Record – Rapid development library for building daemon independent database queries
- »» Action Pack – An implementation of Model View Controller for Ruby.
- »» Action Mailer – An Email Handler
- »» Webserver – Usually webrick, however we use mongrel



Mongrel is the Web Server used for serving Ruby on Rails applications

- »» One instance per Ruby application
- »» Other daemons exist, but mongrel has the best security and performance record
- »» Is a gem
- »» Runs applications on port 12001 and up on cPanel
- »» Uses a significant amount of memory



cPanel and Ruby on Rails

cPanel provides an interface for managing Ruby on Rails applications inside of each user's cPanel

- » Can be enabled/disabled via the feature manager in WHM
- » Number of Ruby applications a user is allowed can be modified via “Modify an Account” in WHM with the “Max Mongrels” option
- » This number should be limited as ruby uses it's own webserver for each application that uses memory




The WHM/cPanel Interface



Create Ruby on Rails Application

App Name *15 character limit





Load on Boot? ☐

Application Path 

Enviroment:

Create

Available Ruby on Rails Applications

APP NAME	PATH	RAILS SERVER	APP STATUS	ACTIONS	LOAD ON BOOT?	PRODUCTION	DELETE
Mephisto	 /Mephisto	URL	Not Running	 Run  Stop	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	



What cPanel Does with RoR

- »» When cPanel creates this application, it runs
“rails /path/to/application”
- »» The information for the application is stored in
~/.cpanel/ruby-on-rails.db
- »» A mongrel process is executed when “run” is hit
- »» each user has their own set of ruby binaries, libraries
and gems in ~/ruby/



Rails and the Shell

Ruby on Rails is designed for each application to be administered from the shell

- » Nearly all applications assume shell access to the hosting server
- » The Rakefile is used to deploy rails application, this will automatically setup databases and many other portions of an application, this is accessed using the “rake” command



How Apache Interacts with RoR

Apache interacts with Ruby on Rails by settings up
Rewrites

- »» Uses `mod_rewrite` in `~/public_html/.htaccess`
- »» stored in `~/cpanel/ruby-on-rails-rewrites.db`
- »» You can only redirect the main domain, subdomain or addon domain to a ruby on rails application.



Rewrite Interface



Manage Rewrites

Since applications are running on a different port than the other URLs on your server, you'll need to redirect incoming traffic to that port. To do this, you can create a URL rewrite to send users to your Ruby on Rails application.

Create A Rewrite

APP NAME	ACTION
Mephisto	<input type="button" value="Create Rewrite"/>

Current Rewrites

APP NAME	REWRITE URL	ACTIONS
No applications have rewrites		



Rewrite Creation Interface

Create a URL Rewrite

Since applications are running on a different port than the other URLs on your server, you'll need to redirect incoming traffic to that port. To do this, you can create a URL rewrite to send users to your Ruby on Rails application.

APP NAME	DOMAIN & URL TO BE REDIRECTED TO RAILS APP (EX. /MYRAILSAPP).	
Mephisto	http:// <div><div>rubyonrails.com</div><div><div>rubyonrails.com</div><div>hogan.rubyonrails.com</div><div>rubyoffrails.com</div><div>rubyoffrails.rubyonrails.com</div><div>** All Public Domains **</div></div></div> /	<div>Save</div>
Requests will be rew	12002/	
y on Rails Applications]		



cPanel and Gems



There are three methods of installing gems on a cPanel system, either via WHM, cPanel or the CLI

- »» The CLI is the recommended method
- »» When installed through cPanel, they are installed to
~/ruby/gems
- »» When installed through WHM or the CLI they are
installed globally to /usr/lib/ruby



cPanel Gem Interface



Find a Ruby Gem

Search or

Install a Ruby Gem

Installed Ruby Gem(s)

MODULE NAME		VERSION	ACTIONS			
alib	0.5.1	<input type="button" value="Update"/>	<input type="button" value="Reinstall"/>	<input type="button" value="Uninstall"/>	<input type="button" value="Show Docs"/>	
hpricot	0.6	<input type="button" value="Update"/>	<input type="button" value="Reinstall"/>	<input type="button" value="Uninstall"/>	<input type="button" value="Show Docs"/>	
mocha	0.5.6	<input type="button" value="Update"/>	<input type="button" value="Reinstall"/>	<input type="button" value="Uninstall"/>	<input type="button" value="Show Docs"/>	
rake	0.8.1	<input type="button" value="Update"/>	<input type="button" value="Reinstall"/>	<input type="button" value="Uninstall"/>	<input type="button" value="Show Docs"/>	
rfacebook	0.9.8	<input type="button" value="Update"/>	<input type="button" value="Reinstall"/>	<input type="button" value="Uninstall"/>	<input type="button" value="Show Docs"/>	



The gem CLI allows you to install, troubleshoot and perform maintenance on Ruby modules

- »» Similar to cpan and pear
- »» Gems are installed to `/usr/lib/ruby/gems/1.8/gems`
- »» For users, they need to run either `/scripts/gemwrapper` or `~/ruby/bin/gem`



gem CLI (cont.)



```
gem list
```

» This will list all of the gems currently installed on your server

```
gem list --remote
```

» will list all gems available from <http://gems.rubyforge.org/>



gem CLI (cont.)



```
gem install <gemname>
```

»» Installs a gem

»» Can be specified with -v for a specific version of a gem (such as rails)

»» useful for testing errors with our gem installation method

gem CLI (cont.)



```
gem check --specification <gemname>
```

» This will allow you to see what files a module will install, what dependencies it has and general information on the gem

```
gem check --alien <gemname>
```

» Security check, will check for any abnormal files inside of the module's base

```
gem check --verify <filename>
```

» Checks the MD5sum of a file from a module versus the repository



gems without the gem command

Ruby modules can be downloaded from <http://gem.rubyforge.org>. This allows gems to be installed even if the gem CLI is not working

You will need to untar the package, chdir into the directory, then run `“ruby setup.rb”`



Logs For Application Startup Issues

`/usr/local/cpanel/logs/error_log`

» This will show any issues with executing the mongrel process itself

`<APPBASE>/log/mongrel.log`

» This will show any issues with mongrel failing to load gems, 90% of failed starts will be contained here



Logs For Application Errors

`<APPBASE>/log/(production |
development).log`

- These logs will show any errors that the application has while executing.
- If your user is complaining about an application acting funny, it will probably be here
- most issues inside of these logs indicate application side errors that cannot be fixed at the server level.



Application Startup Issues

The best way to check application startup issues is to execute the application manually with the following command:

```
chdir <APPBASE> && sudo -u <USERNAME> /  
usr/bin/ruby /usr/bin/mongrel_rails  
start -p <port> -d -e production -P  
log/mongrel.pid
```

» This will execute the mongrel instance on it's own, outside of cPanel making process tracing simple



Gem Installation Issues

Our gem installer is executed via `/scripts/gemwrapper`, which initializes some environment variables into gem that make it work with our homedir setup. We also use a cache file located at:

```
~/.cpanel/datastore/  
_scripts_gemwrapper_--noexpect_list_--  
remote
```

to store a list of these gems, if this file exists but does not contain a list of gems, remove it and refresh the page



Gem Installation Issues (cont)

There are a few other things that should be checked for gem installation errors

- » Does the user have compiler access?
- » Can the user write to their own `~/ruby/` directory?
- » User Quota?



Question & Answer