RVM

Cut Rubies With Ease

Ruby Version Manager

- Multiple Ruby versions (compiled) and installed in parallel
- Each Ruby version has its own environment, lives in different directories
- Shell scripts to manage PATH, GEM_PATH, GEM_HOME and much more

Why?

Ecosystem

- MRI (1.8)
- YARV (1.9)
- REE
- Rubinius
- JRuby
- MacRuby
- IronRuby, MagLev, HotRuby (!)

MRI (1.8)

- Gold Ruby Standard
- Reference implementation (written in C)
- Good enough for **most** applications
- **Slow**: no bytecode, no JIT
- No built-in encoding support - only Iconv
- GC **sucks**

REE (1.8)

- From Passenger (Apache/Nginx module for Rails apps) devs - phusion.nl
- COW friendly
- Google's tcmalloc
- Union Station - released **yesterday**

YARV (1.9)

- Current stable, built upon 1.8 C codebase
- *Everything works out of the box*

```
unless depends_on?(Unmantained::Stuff)
```

- Faster than 1.8: bytecode, JIT (no rt.jar :)
- Strong encoding support
- GC (still) **sucks**

Rubinius

- Ruby deserves to become **first-class** citizen
- Most of the stdlib written in Ruby
- Written in C++, bytecode, JIT
- Built on LLVM, spinned off RubySpec
- FFI subsystem for existing C extensions
- http://rubini.us/2011/02/25/why-use-rubinius/

JRuby

- Ruby built on the JVM - from 1.5 onwards
- Fully **interoperable**
  - Works {on,with} Java {servers, libraries}
  - Call Java from Ruby and Ruby from Java
- FFI subsystem for existing C extensions
- **Fast** performance (JIT), **Slow** startup time

MacRuby

- Sponsored by Apple, works only on OS X
- Built upon Objective-C runtime and LLVM
- A Ruby wrapper around CoreFoundation
- JITted, compiles to binary code
- **Real GUIs** can be built with it

Others

- IronRuby is Ruby on .NET
- MagLev is distributed object persistence amongst networked Ruby processes
- HotRuby is Ruby in the browser via Javascript and Flash **(!)**
- RVM is the easiest way to try them all!

Install RVM

- Run `bash < <(curl http://rvm.beginrescueend.com/releases/rvm-install-head )`
- Add `[[ -s ~/.rvm/scripts/rvm ]] && source ~/.rvm/scripts/rvm` to ~/.bashrc
- Set!

RTFM

http://rvm.beginrescueend.com/

Install something

- `rvm install 1.8` # the last 1.8 release
- `rvm install 1.9` # as above, for 1.9
- `rvm install rbx` # Rubinius
- `rvm install ree`
- `rvm install jruby`

What have we got?

```
$ rvm list
rvm rubies
   macruby-0.6 [ x86_64 ]
   rbx-1.2.2-20110222 [ ]
   ree-1.8.7-2010.01 [ ]
   ruby-1.8.7-p330 [ ]
   ruby-1.9.2-p0 [ ]
```

System ruby

```
$ rvm system
$ type ruby
ruby is /usr/bin/ruby
$ ruby -v
ruby 1.8.7 (2009-06-12
patchlevel 174) [universal-
darwin10.0]
```

Let's switch!

```
# Caveat - exact ruby version
$ rvm 1.9.2p0
$ type ruby
ruby is /Users/vjt/.rvm/rubies/
ruby-1.9.2-p0/bin/ruby
$ ruby -v
ruby 1.9.2p0 (2010-08-18
revision 29036) [x86_64-
darwin10.4.0]
```

Again

```
$ rvm rbx-1.2.2
$ type ruby
ruby is /Users/vjt/.rvm/rubies/
rbx-1.2.2-20110222/bin/ruby
$ ruby -v
rubinius 1.2.2 (1.8.7 release
2011-02-22 JI) [x86_64-apple-
darwin10.5.0]
```

What am I using?

```
$ rvm list

rvm rubies
   macruby-0.6 [ x86_64 ]
=> rbx-1.2.2-20110222 [ ]
   ree-1.8.7-2010.01 [ ]
   ruby-1.8.7-p330 [ ]
   ruby-1.9.2-p0 [ ]
```

Upgrade?

```
# Ruby
$ rvm upgrade 1.9.2-p0 1.9.2-
p180
```

```
# RVM itself
$ rvm get head
```

Gemsets

```
$ rvm 1.9@someapp
$ gem list
*** LOCAL GEMS ***
abstract (1.0.0)
actionmailer (3.0.3)
actionpack (3.0.3)
activemodel (3.0.3)
activerecord (3.0.3)
......
```

Gemsets, continued

```
$ rvm 1.9@antani
$ gem list
*** LOCAL GEMS ***
rake (0.8.7)
$
```

Work in isolation on different apps with different dependencies.

Development

- Put .rvmrc in the root of your project
- Add it to the ignore list of your SCM
- RVM will execute it when you enter the root

```
$ cat .rvmrc
rvm 1.8.7@ifad-members
[ -z "$SYBASE" -a -x /opt/sybase/SYBASE.sh ] &&
{
  echo "loading sybase environment..."
  . /opt/sybase/SYBASE.sh
}
```

Production?

.bashrc:

```
source ~/.rvm/scripts/rvm
```

runner:

```
exec sudo -i -H -u someuser bash
-c 'echo; cd; exec unicorn
 -c config/unicorn.conf.rb
 -E production -D'
```

Production - more

- Keep applications **isolated** from each others
- Run applications requiring different interpreters along each other
- Compile ruby with `-static` and run everything in a `chroot()` [coming soon]
- Different users? Just use symlinks

Thank you

http://sindro.me/

http://twitter.com/vjt

vjt@openssl.it