



JavaOne

JRuby on Rails: Agility for the Enterprise

Thomas E Enebo, Charles O Nutter

JRuby Core Developers
Sun Microsystems, Inc.
<http://www.jruby.org/>

TS-9370

2007 JavaOneSM Conference | Session TS-9370 |

java.sun.com/javaone

Goal of This Talk

What You Will Gain

Get an introduction to Ruby, JRuby, and Ruby on Rails. See how the other 8% lives.

Agenda

Introduction

Ruby

JRuby

Ruby on Rails

JRuby on Rails

Futures

Introductions

- Charles Oliver Nutter and Thomas Enebo
- Engineers at Sun Microsystems
- Full-time JRuby developers
- Over ten years Java technology experience each
- Formerly a Java™ Platform, Enterprise Edition (Java™ EE platform) architect
 - Also C/C++, Win32, C#, .NET, CLDC
- Tom was a web application architect
 - Also C, Perl, Jini™ network technology, Java Platform, Micro Edition (Java ME platform)
- Working to make Ruby a first-class Java Virtual Machine (JVM™) language

The terms “Java Virtual Machine” and “JVM” mean a Virtual Machine for the Java™ platform.

Agenda

Introductions

Ruby

JRuby

Ruby on Rails

JRuby on Rails

Futures

What Is Ruby?

- Dynamically-typed pure object-oriented programming language
 - Focus on simplicity, productivity, and “fun”
 - Interpreted
 - Open source, written in C
 - Implementation is the Specification
- Created 1993 by Yukihiro “Matz” Matsumoto
 - “More powerful than Perl and more OO than Python”
- Very active community, wide range of apps

Ruby Quick Tour: Pure OO

- Pure Object-oriented language
- Absolutely everything is an object
 - `7.class` `=> Fixnum`
 - `"hello".length` `=> 5`
 - `3.times { |i| puts "Number #{i}" }`
`=> Number 1`
`Number 2`
`Number 3`

Ruby Quick Tour: Sweet Sugar

- Common syntactical sugar
 - Regular expressions: `/^foo.*bar/`
 - Arrays: `['one', 1, Ein.new]`
 - Associative arrays: `{ 'a' => 3, 'b' => 4 }`
 - Strings: `"foo"`

Ruby Quick Tour: Dynamic Typing

- Dynamically-typed language
 - No need to declare variables
 - Duck-typing
 - “If it quacks like a duck and waddles like a duck, then ...”

```
def make_it_waddle(waddler)
  waddler.waddle
end
```

```
make_it_waddle(Duck.new)
make_it_waddle(Penguin.new)
make_it_waddle(Octopus.new)
```

Ruby Quick Tour: Modules

- Modules mix-in behavior

```
module ShapeStuff
  def diameter
    return 2 * @radius
  end
end
```

```
class Circle
  include ShapeStuff
  ...
end
Circle.new(4).diameter ==> 8
```

Ruby Quick Tour: Blocks

- Anonymous methods you can pass around and invoke
 - Add block to be invoked on end of method call
 - Removes repetitive code
 - `list.each { |element| ... do something ... }`
 - Internalizes transactional code
 - `File.open(fname) { |file| line = file.gets ... }`
 - Makes nice APIs

Ruby Quick Tour: Metaprogramming

- Meta programming
 - Dynamically generate classes
 - Modify existing classes
 - Add/Alias/Remove/Replace methods
 - Include a module
 - Add instance/class variable
 - Special dispatching to pretend things exist
 - `method_missing`, `const_missing`

Agenda

Introductions

Ruby

JRuby

Ruby on Rails

JRuby on Rails

Futures

What Is JRuby?

- Started in 2002
- Java platform implementation of Ruby language
- Open source, many active contributors
- Aiming for compatibility with current Ruby version
- Integrates with Java technology
 - Call to Ruby from Java technology via Java Specification Request (JSR) 223, BSF, Spring
 - Use Java class files from Ruby (e.g., Script Java)
- Growing set of external projects based on JRuby
 - JRuby-extras (GoldSpike, ActiveRecord-JDBC,...)

JRuby: Java Technology Integration

- Script Java
 - Power of Java technology with the Syntax of Ruby

```
require 'java'
```

```
list = java.util.ArrayList.new
```

```
list << 1
```

```
list << 3
```

```
list << 2
```

```
list.sort
```

Why JRuby?

- JRuby over Ruby
 - Strong likelihood we will be faster soon
 - Better scalability with native threading
 - Native unicode support
 - Compilation
 - Integration with Java libraries
 - Easier path to getting Ruby in the enterprise
- JRuby over Java technology
 - Language features
 - Blocks, modules, metaprogramming, dynamic-typing
 - Ruby applications/libraries
 - Rails, Rspec, Rake, Raven, other R's

Agenda

Introductions

Ruby

JRuby

Ruby on Rails

JRuby on Rails

Futures

What Is Ruby on Rails?

- A full-stack MVC web development framework
- Open source (MIT), many contributors
- Written in Ruby
- Single-threaded design
- First released in 2004 by David Heinemeier Hansson
- Ruby and Rails book sales are outselling Perl book sales
- Shipping with “Leopard”

Ruby on Rails Precepts

- Convention over configuration
 - Why punish the common cases?
 - Encourages standard practices
 - Everything simpler and smaller
- Don't Repeat Yourself (DRY)
 - Framework written around minimizing repetition
 - Repetitive code harmful to adaptability
- Agile development environment
 - No recompile, deploy, restart cycles
 - Simple tools to generate code quickly
 - Testing built into framework

Why Ruby on Rails?

- Greatly simplified web development
 - “Less Rails code than Java application configuration”
 - Instant applications: working code in minutes
- Growing excited community of developers
- Makes small apps trivial to create
- Ruby is an excellent language

Why JRuby on Rails?

- Deployment to Java application servers
- Java technology production environments pervasive
 - Easier to switch framework vs. whole architecture
 - Lower barrier to entry
- Broader, more scalable database support
- Integration with Java technology libs, legacy services
- No need to leave Java technology servers, libraries, reliability

Rails Quick Tour: Project Layout

- All Rails apps have same layout

```
myapp
```

```
+ app
```

```
  + controllers
```

```
  + helpers
```

```
  + models
```

```
  + views
```

```
+ config
```

```
+ db
```

```
+ lib
```

```
+ public
```

```
+ test
```

Rails Quick Tour: Tools

- ‘rails myapp’ creates new application
- ‘ruby script/generate’ generates:
 - Models, views, and controllers
 - Migrations (Ruby DB-agnostic DDL management)
 - Scaffolding (quick CRUD maker)
 - Tests (unit and functional)
- ‘rake’ runs tasks
 - Perform migration
- ‘ruby script/server’ starts application
- Many more...

Rails Quick Tour: Components

- ActiveRecord
 - ORM on steroids
- ActionPack
 - View and controller support
- Actionwebservice
 - Guess!
- Actionmailer
 - Email
- Activesupport
 - Unicode, JSON, miscellaneous helpers



DEMO

Ruby on Rails with JRuby



Agenda

Introductions

Ruby

JRuby

Ruby on Rails

JRuby on Rails

Futures

JRuby on Rails: Java EE Platform

- Pool database connections
- Access any Java Naming and Directory Interface™ (J.N.D.I.) API resource
- Access any Java EE platform TLA:
 - Java Persistence API (JPA)
 - Java Management Extensions (JMX™)
 - Enterprise JavaBeans™ (EJB™)
 - Java Message Service (JMS) API
 - SOAP/WSDL/SOA

JRuby on Rails: Deployment

- Mongrel
- Warfile deployment to:
 - Project GlassFish™ V2
 - Servlet-based app server
- Project GlassFish™ V3
- Go to technical keynote at 1:30

Agenda

Introductions

Ruby

JRuby

Ruby on Rails

JRuby on Rails

Futures

JRuby on Rails: Futures

- Improve Java EE platform support for Rails
 - Rubify Java EE platform APIs
 - Create Rails plug-ins to make installable units
- Use Java technology native equivalents
 - Unicode support
 - XML support
- Port Ruby native libraries
 - RMagick

More Information

- JRuby: www.jruby.org
- Ruby: www.ruby-lang.org
- Rails: www.rubyonrails.org
- NetBeans™ IDE: www.netbeans.org
- Charles's blog: www.headius.blogspot.com
- Tom's blog:
www.bloglines.com/blog/ThomasEEnebo



Q&A

Thank You





JavaOne

JRuby on Rails: Agility for the Enterprise

Thomas E Enebo, Charles O Nutter

Jruby Core Developers
Sun Microsystems, Inc.
<http://www.jruby.org/>

TS-9370

2007 JavaOneSM Conference | Session TS-9370 |

java.sun.com/javaone