



# JavaOne™

[java.sun.com/javaone](http://java.sun.com/javaone)

## JRuby on Rails: Web Development Evolved

Ola Bini  
JRuby Core Developer  
ThoughtWorks

TS-4806



**Learn why Ruby on Rails is taking the world by storm and how JRuby can give you access to this technology right now**



GOAL

Learn why Ruby on Rails is taking the world by **storm** and how JRuby can give you access to this technology right now

Sun Microsystems      43 things      Fancast  
 Dreamhost      Twitter      37signals  
 Valtech      ThoughtWorks      EngineYard      APress  
 JustSayHi      CodeHaus      Glue      Jobster      O'Reilly  
 Oracle<sup>®</sup>      RackSpace      Traffic Broker      Revolution Health  
 Addison Wesley      Trifork

# Agenda

- Introduction
- Ruby
- JRuby
- Ruby on Rails
- JRuby on Rails
- The future

# Agenda

- Introduction
- Ruby
- JRuby
- Ruby on Rails
- JRuby on Rails
- The future

# Introduction

- Ola Bini
- Works for ThoughtWorks Studios
- Full-time JRuby developer
- Java™ platform developer for 10 years
- Ruby developer for 5 years
- Involved in JRuby for 3 years
- *Language geek*
  - *C, C++, C#, Smalltalk, Common Lisp, Scheme, Prolog, etc*
- *Author of “Practical JRuby on Rails” from APress*

# Agenda

- Introduction
- Ruby
- JRuby
- Ruby on Rails
- JRuby on Rails
- The future

# What is Ruby

- Strongly typed, pure object oriented, dynamic language
  - Focus on simplicity, productivity and “fun”
  - Interpreted
  - Open Source, written in C
  - C Implementation == Specification
- Created 1993 by Yukihiro “Matz” Matsumoto
  - “More powerful than Perl, more OO than Python”
  - First released 1995
- Very active community
  - Focus on Open Source
- Wide range of applications



# Ruby features – Object Orientation

- 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 features – 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 features – Syntactic sugar

## ➤ Common syntactic sugar

## ➤ Literals

- Regular expressions: `/^foo.*bar$/, %r[/ /]`
- Arrays: `['one', 1, Ein.new], %w(several words here)`
- Associative Arrays (hashes): `{ 'a' => 3, 'b' => 4 }`
- Strings: `"foo", %[abc foo "bar" is a string]`

## ➤ Punctuation

- *No need for semi-colon, except for more than one thing on a line*
- *Leave off parenthesis when no ambiguity*
- *Choose {...} or do...end for blocks*

# Ruby features – Blocks

## ➤ Anonymous methods you can pass around and invoke

- Add block to be invoked on end of method call
- Closures
- Removes repetition
  - Iteration
- Internalizes transactional code
  - File opening and closing
  - Database transactions
- Make it easy to create nice APIs

## Ruby features – Modules

- Modules are namespaces
- But also allow mixins:

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

```
class Circle
  include ShapeStuff
end
```

```
Circle.new(4).diameter    => 8
```

# Ruby features – Metaprogramming

## ➤ Classes and modules

- Create new ones dynamically
- Mix in new modules at any time

## ➤ Methods

- Create new ones from blocks
- Dynamic naming
- Add, remove, alias or redefine any method at any time
- `method_missing` to fake method existence

## ➤ Instance variables can be dynamically accessed

## ➤ Use `send` to call any method (like Java platform reflection)

# Agenda

- Introduction
- Ruby
- JRuby
- Ruby on Rails
- JRuby on Rails
- The future

# What is JRuby

- Started in 2001
  - Created by Jan Arne Petersen
- Java platform implementation of the Ruby language
- Open Source, many active contributors
  - Currently 8 core developers
  - Some amazing contributions: Marcin's port of Oniguruma
- Fully compatible with Ruby 1.8.6
- Current version 1.1 and 1.0.4
- Growing set of external projects based on JRuby
- Commercial backing
  - Sun Microsystems
  - ThoughtWorks



# Why JRuby

## ➤ JRuby over Ruby

- Performance – JRuby is between 2-5x faster than Ruby for almost all benchmarks
- Better scalability with native threads
- Native Unicode support
- Full compilation ahead-of-time or just-in-time
  - But still retaining interpreted mode
- Integration with Java platform libraries
- Good way to get Ruby into the enterprise

## ➤ JRuby over Java platform

- Missing language features
- Ruby applications/libraries

# Integration with Java Technology

- Access to any Java platform library with intuitive syntax

```
require 'java'
```

```
import java.util.HashMap
```

```
h = HashMap.new
```

```
h["foo"] = "bar"
```

```
h["quux"] = "baz"
```

```
h.key_set.sort.each { |key| puts key }
```

# Integration with Java Technology

- Use Ruby from within a Java program

```
// One-time load Ruby runtime
ScriptEngineManager factory =
    new ScriptEngineManager();

ScriptEngine engine =
    factory.getEngineByName("jruby");

// Evaluate JRuby code from string.
try {
    engine.eval("puts('Hello')");
} catch (ScriptException exception) {
    exception.printStackTrace();
}
```

# Agenda

- Introduction
- Ruby
- JRuby
- **Ruby on Rails**
- JRuby on Rails
- The future

# What is Ruby on Rails

- Full stack MVC web development framework
- Open Source (MIT)
  - Many contributors
- Written in Ruby, utilizing the best language features to the max
- Single-threaded, shared-nothing design
- First released 2004 by David Heinemeier Hansson
- Current version 2.0.2
- Ships with MacOS X 10.5 (Leopard)

# Tenets of Ruby on Rails

## ➤ Convention over configuration

- The common case should be the default
- Encourages standard practices
- Everything simpler and smaller
- But doesn't lock you in, you can configure if you want to

## ➤ Don't Repeat Yourself (DRY)

- The whole framework tries to minimize repetition
- ... since repetitive code is harmful to adaptability and rapid development

## ➤ Agile development

- No recompile, deploy, restart cycles
- Simple tools that generate code quickly
- Testing built into the framework

# Why Ruby on Rails

- Greatly simplified web development
  - “Less Rails code than Java application configuration”
  - Instant applications, working code in minutes
- Excited community of developers
  - Lots of innovation happening around Rails
- Small applications are trivial to create
  - And larger applications bearable
- Ruby is an excellent language and Rails uses this to max

# Rails Quick Tour

- All Rails applications have the same layout:

**myapp**

+ **app**

+ **controllers**

+ **helpers**

+ **models**

+ **views**

+ **layouts**

+ **config**

+ **db**

+ **lib**

+ **public**

+ **test**



# Rails Quick Tour

## ➤ Code generators

- 'rails myapp' creates a directory 'myapp' with basic structure
- 'ruby script/generate' generates
  - Models, views, controllers
  - Migrations
  - Scaffolding
  - Tests
  - ... anything a plugin what like to generate

## ➤ Rake used to run tasks

- Database migrations
- Testing

## ➤ 'ruby script/server' starts the application

# Creating a Rails application with JRuby

A large, light blue, semi-transparent graphic consisting of a chevron pointing right followed by the word "DEMO" in a bold, sans-serif font.

# Agenda

- Introduction
- Ruby
- JRuby
- Ruby on Rails
- JRuby on Rails
- The future

# Why JRuby on Rails

- All the general JRuby reasons
- Deployment to Java application servers
- Java application environments are pervasive
- Broader, more scalable database support with JDBC™ API
  - Hundreds, maybe thousands of drivers
  - Connection pooling
- Integration with Java platform libraries and legacy services

# JRuby on Rails with Java Platform, Enterprise Edition (Java EE platform)

- Pooling of database connections
- Access any Java Naming and Directory Interface (JNDI) API resource
- Access any Java EE platform library:
  - Java Persistence (JPA) API
  - Java Management Extensions (JMX) API
  - Enterprise JavaBeans™ (EJB™) architecture
  - Java Message Service (JMS) API
  - SOAP and WSDL
  - XML libraries

# Ruby on Rails: Traditional deployment

- Apache or NginX
- mod\_proxy or equivalent
- Several Mongrel instances
  
- Works, but is cumbersome
- Doesn't scale to many applications
  - For M Rails applications
    - For N concurrent requests to each app
      - And monitor process for each
        - At least  $M*N$  Mongrels
        - That will die or go Zombie on you
  
- Needs to be heavily maintained

# Ruby on Rails: Deployment with JRuby

- Use many Mongrels inside of one JVM
- Or use the GlassFish™ application server gem
- Or package up into a Web Application Archive (WAR) file
  - And deploy to any compliant Web Container
  - Several tools for packaging:
    - Warbler
    - GoldSpike
    - JRubyWorks

# Deploying a Rails application with JRuby

A large, light blue arrow pointing to the right, positioned behind the word "DEMO".

# DEMO



# JRuby on Rails – Real world usage

## > mix.oracle.com

- Idea sharing
- Networking
- Built in 6 weeks
- Runs on the full Oracle stack

## > mediacast.sun.com

- Media/marketing distribution channel

## > Mingle

- First JRoR product
- Project collaboration tool
- JRuby enables several things not possible with regular Ruby
  - Easy bundling
  - Protection of source code
  - Access to outstanding Java platform libraries

# Agenda

- Introduction
- Ruby
- JRuby
- Ruby on Rails
- JRuby on Rails
- The future

# Future development: JRuby

## ➤ Version 2.0

- Reworked Java platform Integration features
- Public API for extensions
- Better performance
- Support for Ruby 1.9 features
- Light weight objects?
- Support for Java Specification Request 292 (JSR292) `invoke_dynamic`?

# Future development: JRuby on Rails

- Easier and more intuitive packaging
- Support for more databases
- Runtime information sharing to avoid memory hit
- Multithread support in Rails?
- GlassFish application server gem finalized and production ready
- Better integration with Java platform libraries
  - For example: you can use JMX API right now
  - But wouldn't it be neat to have libraries that you can just drop in to your Rails application to enable JMX API?

# Summary

- Ruby is a powerful language
- Ruby together with Java technology is the best of both worlds
- Rails utilizes Ruby to the max
- It's extremely easy to get started with Rails
- JRuby allows adoption of Rails in the enterprise
- JRuby on Rails is arguably the best Web application development on the Java platform

# For More Information

## ➤ Sessions

- TS-5921: GlassFish v3
- PAN-5435: The Script Bowl
- TS-5416: JRuby: Why, What, How... Do It Now
- TS-6490: JRuby on Rails Deployment

## ➤ BOFs

- BOF-4807: JRuby at ThoughtWorks
- BOF-5111: Scripting in GlassFish project

## ➤ URLs

- <http://ruby-lang.org>
- <http://jruby.org>
- <http://rubyonrails.com>

## ➤ Practical JRuby on Rails (APress, 1-59059-881-4)

# THANK YOU

Ola Bini  
JRuby Core Developer  
ThoughtWorks

TS-4806

