

# JRuby

**Charles Oliver Nutter**

**JRuby Guy**

**Sun Microsystems**



Except where otherwise noted, the content of this presentation is licensed under the Creative Commons Attribution-Share Alike 3.0 United States License (<http://creativecommons.org/licenses/by-sa/3.0/us/>).

# Agenda

- JRuby intro
- Tools and IDE support
- Swing programming
- Graphics and applets (that don't suck)
- Web applications
- JRuby future

# Who Am I

- Charles Oliver Nutter
- Longtime Java developer (12+ yrs)
- Engineer at Sun Microsystems for 1 yr
- Full-time JRuby developer
- Also working on JVM dynlang support
- Wide range of past experience
  - > C, C++, C#, Python, Delphi, Lisp, Scheme
  - > Java EE and ME, WS

# JRuby

- Java implementation of Ruby language
  - > “It's just Ruby!”
- Started in 2002, open source, many contributors
  - > Tom Enebo, Ola Bini, Marcin Mielzynsky, Nick Sieger, Vladimir Sizikov, MenTaLguY
- Aiming for compatibility with current Ruby version
  - > Ruby 1.8.6 patchlevel 111 (114 was just released)
- Improvements on Ruby
  - > Native threading, better performance, many libraries

# JRuby 1.1.2

- Maintenance release for 1.1
- Excellent improvements
  - > Drastically reduce raw startup time
  - > YAML symbol parsing >100x faster than 1.1.1
  - > Performance, threading, stack improvements for calls
  - > Several nested backref problems fixed
  - > Bad data race in regexp cache fixed
  - > Gazillions of BigDecimal issues fixed
  - > 95 bugs fixed since 1.1.1

# JRuby 1.1.3 Soon

- 30% faster interpreter
- Multiple compiled perf improvements
- Non-local flow perf fix (break, return, etc in block)
- Dozens more bug fixes
- Several Rails-related bottlenecks removed
- Shared or reduced memory AST (hopefully!)
- Reduced object/GC churn

# Where is JRuby today?

- Tooling for IDEs
  - > JRuby's parser enables NetBeans, Eclipse, IntelliJ
- Swing GUI development
  - > Makes Swing much nicer to use, easier to handle
- Graphics
  - > Ruby + graphics = cool demos
- JRuby on Rails
  - > Better deployment options, better performance

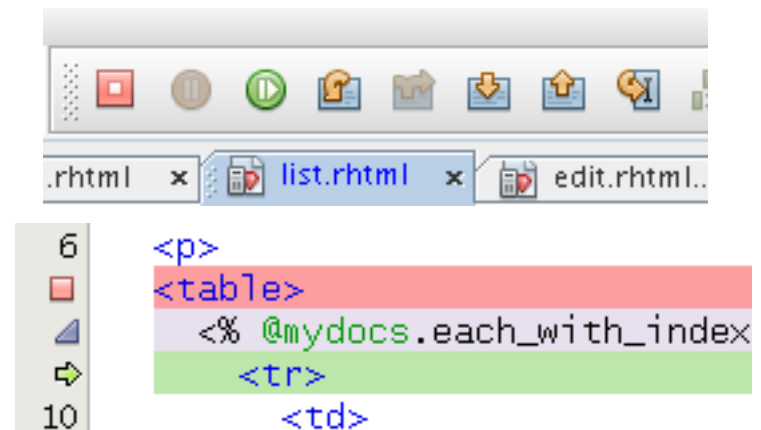
# JRuby Enables Tooling

- JRuby's parser used by most Ruby IDEs
  - > NetBeans Ruby Support
  - > Eclipse RDT/RadRails/Aptana, DLTK, 3<sup>rd</sup> Rail
  - > IntelliJ (not parser, but other areas)
  - > Jedit
- NetBeans is the best right now

```
class Post < ActiveRecord::Migration
  def self.up
    create_table(name, options) ActiveRecord
    create_
  end
end
```

SchemaStatements.create\_table(name, opt: ...)

Creates a new table There are two ways to work with it: use the block form or the regular form, like this:



The screenshot shows a web browser window with three tabs: ".rhtml", "list.rhtml", and "edit.rhtml..". The "list.rhtml" tab is active and displays a list of documents. The list is enclosed in a table structure. The first row is highlighted in red, the second in purple, and the third in green. The table contains a header row and a data row.

```
6 <p>
  <table>
    <% @mydocs.each_with_index
  <tr>
    <td>
10
```

# Swing GUI Programming

- Swing API is very large, complex
  - > Ruby magic simplifies most of the tricky bits
- Java is a very verbose language
  - > Ruby makes Swing actually fun
- No consistent cross-platform GUI library for Ruby
  - > Swing works everywhere Java does (everywhere)
- No fire-and-forget execution
  - > No dependencies: any script works on any JRuby install

# Swing Options

- Cheri - [cheri.rubyforge.org](http://cheri.rubyforge.org)
  - > Builder-like DSL syntax
- Profligacy - [ihate.rubyforge.org/profligacy](http://ihate.rubyforge.org/profligacy)
  - > Rubified layout expression language
  - > Trivial event binding without listeners
- **MonkeyBars - [monkeybars.rubyforge.org](http://monkeybars.rubyforge.org)**
  - > Leverage GUI builders
  - > MVC structure

# **DEMO**

## **Swing in Ruby**

# Graphics Processing

- “Processing is an open source programming language and environment for people who want to program images, animation, and interactions.”
  - > Basically a cool Java library for 2D graphics
- Ruby-Processing wraps Processing with JRuby
  - > Cool, rubified 2D graphics environment for you
  - > Eye-candy demos for us
  - > Thanks to Jeremy Ashkenas for putting these together

**DEMO**  
**A Face for**  
**Stephen Hawking**

# Web applications

- Classic Java web dev is too complicated
  - > Modern frameworks follow Rails' lead
- Over-flexible, over-configured
  - > Conventions trump repetition and configuration
- Rails deployment is still a pain
  - > You shouldn't need N processes!
- Rails performance should be better
  - > JRuby has potential to be much faster

# Deployment: The Old Way

- Mongrel as server process
  - > Times number of apps
    - Times number of concurrent requests
      - With monitoring to keep alive
        - Hopefully no zombies or memory leaks
          - Better have a big slice
  - > ...And then you need to switch hosts/servers
- Sure, it can work...but why?
  - > These problems have been solved before
  - > It's painful because it's wrong!
    - Mongrel is a ghetto?
- (But yes, Mongrel works on JRuby)

# Deployment Improved

- Phusion Passenger + Ruby EE
  - > Apache mod, tweaked Ruby
  - > Improved memory, perf (10-20%?)
  - > Easier deploy: config Apache and done

# A Classic Revisited

- Java app server deployment (WAR files)
  - > N apps
  - > N' concurrent requests
  - > N'' database connections
  - > 1 process
    - Plus full-site management, monitoring, profiling, debugging...
- Designed to scale up and out
  - > ...but at a cost of some agility
- Political standard
  - > Meet them halfway!

# GlassFish Setup

- Download GlassFish from [glassfish.dev.java.net](http://glassfish.dev.java.net)
- `java -Xmx512m -jar glassfish-xxx.jar`
- `ant -f setup.xml (or setup-cluster.xml)`
- `glassfish/bin/asadmin start-domain`

# **DEMO**

## **Rails on GlassFish**





# GlassFish V3 Gem (in dev)

- `gem install glassfish`
- `glassfish_rails <app dir>`

# Best of Both Worlds

- Agile development
  - > CLI support, dev time server, no build/deploy
- Simple to use
  - > One-shot execution, both prod and dev
- Scaling, both technical and logistical
  - > Single process for all apps and requests
- Ruby friendly
  - > gem install and go!

# **DEMO**

# **GlassFish Gem**

# Production JRuby on Rails

- CSI's Disease Surveillance System
  - > [www.govtech.com/gt/323349](http://www.govtech.com/gt/323349)
- Oracle's Mix - digg-like social customer site
  - > [mix.oracle.com](http://mix.oracle.com)
- Sun's MediaCast - file distribution portal
  - > [mediacast.sun.com](http://mediacast.sun.com)
- ThoughtWorks' Mingle - collaborative project mgmt
  - > [mingle.thoughtworks.com](http://mingle.thoughtworks.com)
- More on the way!

# Thank you!

- JRuby - [www.jruby.org](http://www.jruby.org)
  - > [wiki.jruby.org](http://wiki.jruby.org)
- GlassFish - [glassfish.dev.java.net](http://glassfish.dev.java.net)
  - > `gem install glassfish`
  - > Looking for bug reports, feature requests!
- NetBeans - [www.netbeans.org](http://www.netbeans.org)
  - > Try it out, send feedback!
- Ruby-processing (google it)
- Charlie's blog: [headius.blogspot.com](http://headius.blogspot.com)