



the
POWER
of
JAVA™



JRuby: Bringing Ruby to the JVM™

Thomas E. Enebo
Aandtech Inc.

Charles Oliver Nutter
Ventera Corp
<http://www.jruby.org>

TS-3059

JRuby Presentation Goal

Learn what JRuby is and how Ruby and JRuby will improve your Java world

Agenda

What Are Ruby and JRuby?

Ruby Features

JRuby Features

Demonstrations

JRuby in the Wild

JRuby's Future

Conclusion

Agenda

What Are Ruby and JRuby?

Ruby Features

JRuby Features

Demonstrations

JRuby in the Wild

JRuby's Future

Conclusion

What Is Ruby?

- Pure Object-Oriented Dynamically-typed Interpreted Language
- Open Source
- C Implementation is the “standard”
- Started in 1993 by Yukihiro ‘Matz’ Matsumoto
 - More powerful than Perl and more OO than Python
 - Guided by principle of least surprise
- www.ruby-lang.org

What Is JRuby?

- 100% Java™ implementation of Ruby
- Open Source, GPL/LGPL/CPL licensed
- Not yet a Ruby-to-bytecode compiler
- Native-threaded
- Runs on Java SE versions 1.4.2 or higher
- Started in Fall of 2001 based on Ruby 1.6
- <http://www.jruby.org>

Java Programming Language and Ruby Compared

```
public class Filter {
    public static void main(String[] args) {
        List list = new java.util.ArrayList();
        list.add("Tim"); list.add("Ike"); list.add("Tina");
        Filter filter = new Filter();
        for (String item : filter.filterLongerThan(list, 3)) {
            System.out.println( item );
        }
    }
    public List filterLongerThan(List list, int length) {
        List result = new ArrayList();
        for (String item : list) {
            if (item.length() <= length) { result.add( item ); }
        }
        return result;
    }
}
```

Java Programming Language and Ruby Compared

Ruby!

```
list = ['Tim', 'Ike', 'Tina']  
list.select {|n| n.length > 3}.each {|n| puts n}
```

```
=> 'Tina'
```


Agenda

What Are Ruby and JRuby?

Ruby Features

JRuby Features

Demonstrations

JRuby in the Wild

JRuby's Future

Conclusion

Ruby Features (Blocks)

- Allows passing code around

- ```
def add_ten(base)
 yield(base + 10)
end
```

- ```
add_ten(5) { |num| puts num } => 15
```

- ```
add_ten(5) { |num| puts to_roman(num) } => XV
```

- Iteration done right

- ```
sum = 0;
```

- ```
collection = [1,2,4,3]
```

- ```
collection.each { |i| sum = sum + i }
```

- ```
puts sum => 10
```

# Ruby Features (Duck-Typing)

- Type checking done at Runtime

```
class SlowList
 def find(criteria) ... end
end
class Tree
 def find(criteria) ... end
end

def search_with(search_type, criteria)
 search_type.find(criteria)
end

search_with(SlowList.new)
search_with(Tree.new)
```

# Ruby Features (Open Definitions)

- Never too late to add methods to a class

- ```
class Fixnum
  def prime?
    ...
  end
end
puts 13.prime?    => true
```

- Or even change an existing one

- ```
class OnlyOnce
 def calc
 def calc; @cachedAnswer; end
 @cachedAnswer = expensive_one_time_calc()
 end
end
```

# Ruby Features (Modules)

- Modules provide a namespace

```
module Color
 RED = [255,0,0];
end
puts Color::RED => [255,0,0]
```

- Modules provide mix-in inheritance

```
module Enumerable
 def sort; each {|i| ..sort logic..}; end
end
class Foo
 include Enumerable
 def each; ..yield all elements of Foo to block..; end
end
puts Foo.new.sort => sorted foo
```

# Agenda

What Are Ruby and JRuby?

Ruby Features

**JRuby Features**

Demonstrations

JRuby in the Wild

JRuby's Future

Conclusion

# JRuby Features (Java Technology in Ruby)

- Import Java classes into Ruby

```
require 'java'
include_class "java.util.Random"
puts Random.new.nextInt() => 1064445193
```

```
include_class "java.lang.System"
System.out.println("bar") => bar
```

```
include_class('java.lang.String') { |p, name|
 "J#{name}" }
JString.new('heh')
```

# JRuby Features (Rubify Java Code)

- Automatic mapping of core types
  - Ruby Fixnum, Array, Hash, String... == long, Java List, Map, String...

```
include_class "java.util.Random"
puts Random.new.nextInt() % 10 => 5
```
- Rubified method name aliases
  - ```
include_class 'java.awt.Color'
puts Color::RED.getBlue => 0
puts Color::RED.blue => 0
puts Color::RED.get_blue => 0
```


JRuby Features (Rubify Java Code Cont.)

- Common Ruby methods added to core types
 - `java.lang.Comparable` defines '`<=>`' and includes Ruby's `Comparable` module
 - `java.util.List` defines '`each`', '`<<`' and includes `Enumerable` module
 - `java.util.Map`, `java.util.Set` define '`each`'
 - This list grows over time as we discover good fits

JRuby Features (Java Code ↔ Ruby)

- Implement Java interfaces from Ruby

```
include_class "com.foo.MyInterface"  
class MyImpl < MyInterface  
  def bar # impl of public String bar();  
    "hello"  
  end  
end
```

- Use Ruby from Java code

```
MyInterface mine =  
  (MyInterface) BSF.eval("MyImpl.new");  
mine.bar(); => "hello"
```

JRuby Features (Miscellaneous)

- Runs with Ruby's standard libraries
- Native threading: Ruby thread == Java thread

Agenda

What Are Ruby and JRuby?

Ruby Features

JRuby Features

Demonstrations

JRuby in the Wild

JRuby's Future

Conclusion

DEMO

Interactive JRuby



DEMO

JRuby on Rails!

Agenda

What Are Ruby and JRuby?

Ruby Features

JRuby Features

Demonstrations

JRuby in the Wild

JRuby's Future

Conclusion

JRuby in the Wild

- RDT
 - Ruby Development Tools for Eclipse
 - Uses JRuby's Ruby parser and AST
 - www.rubypeople.org
- Rad Rails
 - RDT-based Ruby on Rails IDE
 - www.radrails.org
- JEdit
 - A Programmer's Text Editor
 - Uses JRuby's Ruby parser and AST
 - www.jedit.org
- DataVision
 - Open Source Report Writer
 - Ruby as default formula language
 - datavision.sourceforge.net

Agenda

What Are Ruby and JRuby?

Ruby Features

JRuby Features

Demonstrations

JRuby in the Wild

JRuby's Future

Conclusion

New JRuby Design

- Iterative interpreter engine
- Heap-allocated Ruby stack frames, scopes
- M:N threading with thread, IO schedulers
- Compiled methods will “trampoline”
- Mixed-mode like HotSpot; dynamic optimization
- Pluggable core modules and subsystems
- Integrated Java technology/Ruby online debugging

JRuby's Future

- Short Term
 - Better compatibility with Ruby 1.8
 - Expanding application support
 - Continuation support
 - Optimizations for current interpreter (Fast)
- Medium Term
 - Incorporating Ruby 2.0 features
 - M:N threading
 - Multi-VM support
 - “JRuby Bytecode” interpreter (Faster)
- Long Term
 - Compilation to Java Bytecode (Fastest!)
 - Tail-Call optimization

Challenges for the Future

- Performance must be improved
 - Much slower than C Ruby currently
 - ...but not many optimizations yet
 - ...and no compilation yet
 - “Fast enough” is not always fast enough
- Keep JRuby Working
 - Refactored and redesigned code must not regress
 - Existing library of tests helps avoid regression
 - Slow process of encapsulating, reimplementing
 - Much more challenging

Challenges (Cont.)

- JVM and Ruby Incompatibilities
 - Continuations require stack manipulation (longjmp)
 - Ruby's Threads allow stop, kill, critical sections
 - Bindings, eval allow executing code in other contexts
 - Class, module defs always open, always mutable
 - System calls, signals, fork, symlinks, etc. not possible
- JRuby isn't our day job
 - Contributors help immensely
 - Always more to do than time available

Agenda

What Are Ruby and JRuby?

Ruby Features

JRuby Features

Demonstrations

JRuby in the Wild

JRuby's Future

Conclusion

Conclusion

- Ruby is an elegant, powerful language
- JRuby gives to Ruby Java technology's capabilities
- JRuby gives to Java technology Ruby's capabilities
- Another tool for the toolbox
- Java VM, like .NET CLR, can support many languages
- Ready for use today, and great things in future

For More Information

- Project Homepage: www.jruby.org
- Ruby Homepage: www.ruby-lang.org
- Charles's Blog: headius.blogspot.com
- Tom's Blog:
www.bloglines.com/blog/ThomasEEnebo
- **Programming Ruby**, by Dave Thomas
- jruby-user and jruby-devel mailing lists

JRuby Q&A

Thomas E. Enebo, Aandtech Inc.

Charles Oliver Nutter, Ventera Corp



the
POWER
of
JAVA™



JRuby: Bringing Ruby to the JVM™

Thomas E. Enebo
Aandtech Inc.

Charles Oliver Nutter
Ventura Corp
<http://www.jruby.org>

TS-3059