

概要

JRubyとは

JVM上で動くRuby実行系
Rubyと同じ文法
Bean Scripting Framework(BSF)サポート
0.8.2(Ruby 1.8.2相当)が公開
CPL/GPL/LGPLでトリプルライセンス

パッケージ内容(bin版)

jruby-0.8.2

- bin: 実行シェル
- docs: ドキュメント
- lib: クラスアーカイブ(Jar)
- samples: サンプルソース

特徴(長所)

JavaからRubyを呼べる
RubyからJavaが呼べる

特徴(短所)

他のScript実装系よりかなり遅い
Processがない, ThreadはRubyのソレと違う
一部ライブラリが動かない(Nativeモノとか)

その他

Java APIの呼び出しが0.8.2系はあまり速くないが, 0.8.3系は速くなるらしい
0.8.4系は, Railsが動く予定らしい

デモ

Hello, world! ()

sample1.rb

```
def distance(x, y)
  Math::sqrt(x * x + y * y)
end

def position()
  [rand * 2.0 - 1.0, rand * 2.0 - 1.0]
end
```

```

def pi(count)
  delta_area = 4.0 / count
  area = 0.0
  for i in 1..count
    x, y = position
    if distance(x, y) <= 1.0
      area += delta_area
    end
  end
  area
end

count = 10
for i in 2..6
  puts "10^{i} : #{pi(count)}"
  count *= 10
end

```

JavaからJRuby

Sample2.java

```

import org.apache.bsf.BSFException;
import org.apache.bsf.BSFManager;

public class Sample2 {

  static String source =
    "if /[0-9]+\)\)\([0-9]+\)\)\([0-9]+\)/ === '192.168.0.1'\n"
+ "  f = true\n"
+ "  Regexp.last_match.to_a.each {|no|\n"
+ "    if f then f = false else puts no end\n"
+ "  }\n"
+ "end";

  public static void main(String[] args) {
    BSFManager manager = new BSFManager();
    manager.registerScriptingEngine(
      "ruby",
      "org.jruby.javasupport.bsf.JRubyEngine",
      new String[] { "rb" });
    try {
      manager.eval("ruby", "(java)", 1, 1, source);
    } catch (BSFException e) {
      System.err.println(e);
    }
  }
}

```

JRubyからJava

swing1.rb

```

require 'java'

include_class "javax.swing.JFrame"
include_class "javax.swing.JLabel"

frame = JFrame.new("Hello Swing")
label = JLabel.new("Hello World")

frame.getContentPane().add(label)
frame.setDefaultCloseOperation(JFrame::EXIT_ON_CLOSE)
frame.pack()
frame.setVisible(true)

```

swing2.rb

```

# Import Java packages
require 'java'

```

```
include_class "java.awt.event.ActionListener"
include_class [
  "JButton", "JFrame", "JLabel", "JOptionPane"].map {|e| "javax.swing." + e}

frame = JFrame.new("Hello Swing")
button = JButton.new("Klick Me!")

class ClickAction < ActionListener
  def actionPerformed(evt)
    JOptionPane.showMessageDialog(
      nil,
      "<html>Hello from <b><u>JRuby</u></b>."
      + "<br> Button '#{evt.getActionCommand()}' clicked."
    )
  end
end

button.addActionListener(ClickAction.new)

# Add the button to the frame
frame.getContentPane().add(button)

# Show frame
frame.setDefaultCloseOperation(JFrame::EXIT_ON_CLOSE)
frame.pack()
frame.setVisible(true)
```

まとめ

結局何がうれしいの?

- 膨大なJava APIを呼び出せる
- LLなRubyをJavaから利用できる

おわり

JRuby Home

- <http://jruby.sourceforge.net/>