

English

Yet Another Java Bridge

yajb は pure Ruby, pure Java による Ruby/Java 連携実装です。コンパイル無しで動くので、多くの環境ですぐに使えます。個人的にはクロスプラットフォーム GUI ツールキットや、groovy などのような Java のスクリプト言語という位置づけで開発しています。

サンプルと機能など

Hello world 的

```
require 'yajb/jbridge'
include JavaBridge
jimport "javax.swing.*"

:JOptionPane.jclass.showMessageDialog( nil, "Hello World!")
```

GUI

普通に Java で GUI をするようにプログラムできます。

- [gui_simple.rb](#) : ボタンとイベント (java.awt.event の実装)
- [gui_dialog.rb](#) : モーダルダイアログのデモ (GUI のイベントスレッドが普通に動く例)
- [gui_table.rb](#) : JTable のデモ (TableModel を実装して単純に表示するだけ)
- [gui_table2.rb](#) : JTable のデモ 2 (MVC の例。TableCellRenderer もオーバーライド。)
- [gui_graphics.rb](#) : 描画のデモ (JComponent を実装して絵を描きます)

Java の実装が頑張ってくれているので、日本語の入出力がちゃんとできます。

上のサンプルの画面例

- [gui_dialog.rb](#)

http://www.cmt.phys.kyushu-u.ac.jp/~M.Sakurai/java/ruby/gui_dialog.png

- [gui_graphics.rb](#)

http://www.cmt.phys.kyushu-u.ac.jp/~M.Sakurai/java/ruby/gui_graphics.png

- [gui_table2.rb](#)

http://www.cmt.phys.kyushu-u.ac.jp/~M.Sakurai/java/ruby/gui_table2.png

POI とか JMF とか

[Jakarta-POI](#) は Excel ファイルを操作できますが、これを使うと Linux 上からでも Ruby から Excel が出力できます。(既に Ruby-POI がありますが、Java の機能を使う例として。)

- [ext_poi.rb](#) : POI のデモ (UTF-8)

[Java Media Framework](#) は Java で動画やストリーミングを扱うライブラリですが、これを使うと Ruby でアニメーションを表示したり、動画や静止画をキャプチャーすることができます。

- [jmf_capture.rb](#) : JMF キャプチャーのデモ

http://www.cmt.phys.kyushu-u.ac.jp/~M.Sakurai/java/ruby/jmf_window.jpg
実行時の様子。

ほかにも、Java にあって Ruby に無い機能を Ruby から利用することができます。例 : JDBC ドライバ、JAI、Java3D、JavaMail、Relaxer などなど。

[jlambd](#)、動的 Java プログラミング

内部で Javassist を使っていますので、yajb 上から Javassist を使えます。Javassist を使うことで Ruby 上で動的に Java のクラスを生成することができます。Java のコード片を lambda のように使えるように [jlambd](#) という仕組みもあります。これを使えば速度や微妙な型情報が必要な部分を何とかできます。

- [jlambd.rb](#) : [jlambd](#) デモ

```
p1 = jlambd("int i", "return ($w)(i*2);")
p (1..5).map(&p1)
=> [2, 4, 6, 8, 10]
```

ここで、[jlambd](#) の行は

```
p1 = jlambd("int i", "return new Integer(i*2);")
```

と同じで、([\\$w](#)) はプリミティブ型を自動変換する Javassist の記号です。もちろん、[p1](#) の意味は

```
p1 = lambda {|i| i*2}
```

です。

Javassist については [Javassist Home Page](#) を参照してください。

ドキュメント

[README.ja](#) にある程度まとまった内容を書きました。

- [README.ja](#)

あとは、[jbridge.rb](#) に多少 [rdoc](#) で書いてあります。

- [RDoc API document](#)

必要環境

- ・ Ruby 1.8.2 程度 (1.8.2 でチェックしています)
- ・ Java2SE 1.4 以上の Java 実行環境

ダウンロード

ソースからコンパイルするには ant と JUnit が必要です。

0.8.1 (2005-0906)

- ・ [source and binary](#) (1.9Mb)
- ・ 変更
 - ・ バグ修正 : 標準出力周り (thanks > aivanov)
 - ・ 標準出力の 2 行目以降を出力できないバグの修正
 - ・ 標準出力を表示しない設定の場合、バッファがいっぱいになって Java の処理が止まってしまうバグの修正
 - ・ 標準出力はデフォルトで表示するようにした (表示する方が問題の解決が早い)

0.8.0 (2005-0811)

- ・ [source and binary](#) (1.8Mb)
- ・ 変更
 - ・ バグ修正
 - ・ [] とか [nil] などの配列を渡すとエラー
 - ・ float 型の引数を持つメソッドを検索できない
 - ・ 基本型のクラスを扱えない
 - ・ irb で困らないように、JObject#inspect を見やすくした。
 - ・ Java の標準出力を表示できるようにした。
 - ・ ただし、Windows では速度が極端に低下したり、動かなかったりする場合があります。

0.7.0 (2005-0809)

- ・ [source and binary](#) (1.8Mb)
- ・ 変更
 - ・ メソッド検索ロジックを再実装 (以前の版では、メソッドの検索が Class#getDeclaredMethods から返ってきた配列の順序に依存していました。つまり、最初に一致したメソッドを返すようになっていました。今回の版から、一番もっともらしいメソッドを検索するようになりました。)
 - ・ test コードをパッケージに含めるようにした

0.6.0 (2005-0804)

- ・ [source and binary](#) (1.8Mb)
- ・ 変更
 - ・ irb で Java のメソッド名を補完できるようにした。
 - ・ ブロックで Java のメソッドを実装できるようにした。

- バグ修正をいくつか (cygwin の ruby で動かないとか、float や配列の送受信の問題など)
- ドキュメントを追加
- ライセンスを LGPL に変更

0.5.0 (2005-0719)

- [source and binary](#) (1.8Mb)
- 変更
 - :bstream ドライバを改善
 - 生成したオブジェクトが GC できるようにした
 - jlambd を追加
 - 小バグ修正をいくつか

0.4.0 (2005-0708)

- [source and binary](#) (1.8Mb)
- 変更
 - :bstream を追加。
 - 小バグをたくさん修正。

0.3.0 (2005-0622)

- [binary](#) (1.6Mb)
- [source](#) (1.6Mb)

インストール方法

展開後、コマンドラインで以下のように打ち込んでください。
UNIX 系 OS ではおそらく root 権限が必要になります。

```
($ su)
# ruby setup.rb
```

ライセンス

yajb 自体は LGPL に従って配布します。

XercesJ, xmlrpc は Apache Software License に従って配布します。

This product includes software developed by the
Apache Software Foundation (<http://www.apache.org/>).

Javassist は千葉滋氏の著作物で MPL に従います。

- <http://www.csg.is.titech.ac.jp/~chiba/javassist/>