

Japanese

Yet Another Java Bridge

This module provides a communication facility between Ruby and Java.

The current implementation is written by pure Java and pure ruby so as to run on many platforms.

Simple example and features

These sample files are included in the package file.

Hello world

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

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

(this Java window sometimes appeared behind the other window.)

GUI

You can use Java GUI components as you use on Java.

See sample scripts:

- [gui_simple.rb](#) : Simple GUI (Implementation of java.awt.event interfaces)
- [gui_dialog.rb](#) : JDialog Demo (Handling the Java-AWT thread)
- [gui_table.rb](#) : JTable Demo (Implementation of simple TableModel)
- [gui_table2.rb](#) : JTable Demo2 (Implementation of MVC and CellRenderer)
- [gui_graphics.rb](#) : Drawing graphics (Implementing JComponent and drawing graphics)

Writing scripts in UTF-8, you can use multibyte code correctly.

Screenshots

- [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

An Excel file can be create by [Jakarta-POI](#).

Before running this sample script, you must get the POI.

- [ext_poi.rb](#) : POI Demo (Using external classpath and multibyte handling)

This POI sample script demonstrates classpath configuration and multibyte-character handling.

jlambda (dynamic java programming)

This package contains the javassist, the byte engineering tool library.

You can make implementation from scratch.

- [jlambda.rb](#) : jlambda Demo (jlambda programming sample)

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

The p1 is equivalent to `p1 = jlambda("int i", "return new Integer(i*2);")`.

Of course, p1 can be written by `{|i| i*2}` in Ruby.

See javassist website, <http://www.csg.is.titech.ac.jp/~chiba/javassist/index.html>

Documents

Please read rdoc documents and the jbridge.rb source code.

- [RDoc API document](#)

Download

Compiling the Java source files, you must get Ant(build tool) and JUnit(unit test tool).

0.8.1 (2005-0907)

- [source and binary](#) (1.9Mb)
- change log
 - Bug fixed around the STDOUT transfer. (thanks > aivanov)
 - Can not show STDOUT from JVM more than two lines.
 - Java stalls because the stdout buffer between java and ruby is full.
 - The default option JBRIDGE_OPTIONS[:jvm_stdout] is not nil from this version.

0.8.0 (2005-0811)

- [source and binary](#) (1.8Mb)
- change log

- Bug fixed:
 - :bstream driver can't send the empty array and the array that consists of nil.
 - the new logic of method searching can't find the methods with conversion of losing precision.
 - jbridge can't treat primitive types.
- Improved JObject#inspect output.
- Added the option of showing STDOUT lines from JVM.

0.7.0 (2005-0809)

- [source and binary](#) (1.8Mb)
- change log
 - re-implemented method searching logic (in the previous releases, the method and constructor searching depended on Class#getDeclaredMethods order.)
 - included test script.

0.6.0 (2005-0804)

- [source and binary](#) (1.8Mb)
- change log
 - modified Object#methods for irb completion.
 - added the implementation by block. (See sample scripts.)
 - some bug fixed (wrong classpath separator for ruby on cygwin and float data handling)
 - append some rdoc documents
 - switched to LGPL.

0.5.0 (2005-0719)

- [source and binary](#) (1.8Mb)
- change log
 - improved :bstream driver
 - implemented GC.
 - added jlambda utility.
 - fixed many bugs.

0.4.0 (2005-0708)

- [source and binary](#) (1.8Mb)
- change log
 - added fast :bstream driver.
 - fixed many bugs.

0.3.0 (2005-0622)

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

Install and update

De-compress archive and enter its top directory.

Then type:

```
# ruby setup.rb
```