

Releasing the ooRexx-Java Bridge BSF4ooRexx850

The Bean Scripting Framework for ooRexx

Exploiting all of Java from ooRexx

The 2024 International Rexx Symposium
Brisbane, Queensland, Australia
March 3rd – March 6th 2024



Overview



- Some general remarks on Java
- Brief description of BSF4ooRexx850 with an example
- The most important changes and additions to BSF4ooRexx850
- Download links
- Roundup



- Programming language with the following notable features
 - Compiles to machine instructions ("*bytecode*") of an *artificial processor*
 - Needs a "Java virtual machine (JVM)" to execute the bytecodes
 - JVMs are available for all important operating systems and hardware architectures
 - *Hence, a Java class or a Java program, once compiled can be run everywhere!*
 - Distributed with a (huge) "Java runtime environment (JRE)"
 - A *huge Java class library* that offers everything that an application may possibly need
 - E.g. Socket classes for Internet programming, GUI classes for graphical user interfaces, ...
 - Uncountable third party Java class libraries, most available as open-source (e.g. ASF)
 - Most important programs get programmed with Java (even Android applications!)
 - Many professional applications that are not programmed in Java offer Java APIs
 - E.g. SAP, OpenOffice/LibreOffice, ...
- Hence Java is truly a programmer's "treasure trove" for all operating systems!

- External Rexx function package
 - Allows to interact with the Java runtime environment (JRE)
 - Exploit functionality of Java classes
 - Exploit functionality of Java objects
 - ooRexx 5.0 or later, Java 8 or later
 - Package "BSF.CLS"
 - Camouflages Java as ooRexx (Java appears to be dynamic and message based)
 - Supplies class BSF and public routines
- "Everything that is available in Java becomes directly available to ooRexx !"
 - Java: "write once, run everywhere!"
 - Windows, MacOS, Linux, ...

BSF4ooRexx: An Example



```
dim=.bsf~new("java.awt.Dimension", 100, 200)  
say dim~toString
```

```
::requires BSF.CLS      -- get Java support
```

Output:

```
java.awt.Dimension[width=100,height=200]
```

New Baselines for ooRexx and Java



- BSF4ooRexx has as its baselines ooRexx 4.1 or later and Java 6 or later
 - Deployed applications can be run unchanged “forever”
 - Therefore even older installations are able to exploit the ooRexx-Java bridge
 - However: newer features in ooRexx and Java cannot be exploited by BSF4ooRexx
- BSF4ooRexx850
 - New baselines: ooRexx 5.0 or later and Java 8.0 or later
 - Makes it possible to exploit those features that have therefore become available
 - The package name now carries these baseline version numbers to help to clearly distinguish BSF4ooRexx850 from BSF4ooRexxx



Improve Support for Java Rexx Command Handlers



- Rexx command handlers are usually implemented in Assembler, C, C++
- BSF4ooRexx850 makes it easy for Java programmers to create
 - Proper Rexx command handlers in Java
 - Exploit ooRexx 5's redirection abilities for Rexx issued commands from Java
- JDOR (Java2D for ooRexx)
 - From a proof-of-concept to a full implementation (completed)
 - Distributed with BSF4ooRexx850 and therefore immediately available
 - Much easier to use all of Java2D from ooRexx programs than from Java!



New Rexx Command Handler Samples



- Cf. `BSF4ooRexx850/samples/Java/handlers/commandHandlers`
 - Demonstrate how to implement Rexx command handlers in Java
 - Demonstrate how to preconfigure a Rexx interpreter instance with Java implemented Rexx command handlers and run Rexx scripts that use them
 - Demonstrate that it is even possible to implement a Rexx command handler using ooRexx itself (instead of Assembler, C, C++, Java)! :)
 - Makes it possible for Rexx/ooRexx programmers to create proper Rexx command handlers in pure ooRexx
 - Can be useful for creating Rexx command handler adapters for testing or creating compatibility handlers for existing Rexx programs



Improved Support for Java Programmers



- BSF4ooRexx allows Java programs to interact with ooRexx objects
 - Makes it possible to send from Java messages to ooRexx objects
 - The external Rexx function `BSFCreateRexxProxy()` makes the supplied ooRexx object available as a Java proxy enabling Java to interact with it by sending ooRexx messages to the ooRexx object
 - Used e.g. for callbacks from Java to the contained/boxed ooRexx object
 - Makes it possible to implement Java methods in ooRexx that will get invoked by Java
 - ... and much more
- ooRexx 5 introduced the new API `sendMessageScoped()`
 - The Java `RexxProxy` class got enhanced to offer this API to Java programmers as well



New Configuration Directory Layout, 1



- If using Java archives from the Internet one needs to make them available to Java via the **CLASSPATH** environment variable
- Rexx users need not have any Java programming skills to use Java
 - One cannot expect a Rexx programmer to know about **CLASSPATH**
 - Makes it difficult if not impossible to properly set up **CLASSPATH** for them
- BSF4ooRexx850 now has two new directories
 - Installation directory: *BSF4ooRexx850/lib*
 - User directory: *\$HOME/BSF4ooRexx850/lib* (Unix) or *%USERPROFILE%/BSF4ooRexx850/lib* (Windows)
 - This directory must be created manually by the Rexx user
 - Any Java archive that is placed in one of these two **lib** directories will be made available to Java, such that the Java resources therein can be found by Java

New Configuration Directory Layout, 2



- Simply copy the desired jar-file into one of the two **lib** directories
- Ad Installation directory: *BSF4ooRexx850/lib*
 - Whenever a new version of BSF4ooRexx850 gets installed the prior installation directory will get removed, including all Java archive files in the **lib** directory
 - After installing a new version of BSF4ooRexx850 one needs to get the Java archives that the REXX user had added in addition to BSF4ooRexx850
- Ad user directory: *\$HOME/BSF4ooRexx850/lib* (Unix) or *%USERPROFILE%\BSF4ooRexx850\lib* (Windows)
 - All Java archives in this **lib** directory will never be deleted by the BSF4ooRexx850 uninstall script and therefore “survive” an update of BSF4ooRexx850
 - Please note
 - Over time one may want to replace Java archives with newer versions
 - Make sure that there are no duplicate Java archives in either **lib** directory!



New Support for `JDK_JAVA_OPTIONS`



- BSF4ooRexx(850) honors the environment variable `BSF4Rexx_JavaStartupOptions`
 - Allows to supply arguments to Java for configuring the JVM (Java virtual machine)
 - Now will issue an informal message when honoring it to ease debugging
- Java/OpenJDK has a comparable environment variable named `JDK_JAVA_OPTIONS`
 - Allows to supply arguments to Java for configuring the JVM (Java virtual machine)
 - Now will honor this environment variable and issue an informal message when honoring it to ease debugging



New Support for `JAVA_HOME`, 1



- The environment variable `JAVA_HOME` has become a widely used environment variable in the Java world
 - It points to the directory of a specific Java/OpenJDK directory
 - Makes it possible to have multiple Java/OpenJDK directories (for different versions of Java/OpenJDK) and determine which one should be used
 - Allows for executing different Java/OpenJDK installations in different sessions in parallel
- There is a new launch script named `rexxjh.sh/rexxjh.cmd`
 - Will honor `JAVA_HOME` and use that particular version of Java/OpenJDK to run the supplied REXX script
 - Great to test a REXX program against different versions of Java/OpenJDK while developing or testing
 - Allows for using a specific Java/OpenJDK version, if need be



New Support for **JAVA_HOME**, 2



- If Java gets loaded via the native BSF4ooRexx850 library, **JAVA_HOME** will get honored in the following manner
 - Unix
 - If /opt/BSF4ooRexx850/libjvm.{dylib|so} is present (can be deleted)
 - If present determines which Java/OpenJDK must be used
 - If not present, then the following sequence of Java/OpenJDK locations will be used for loading
 - \$JAVA_HOME/lib/server/libjvm.{dylib|so}
 - \$JAVA_HOME/jre/lib/server/libjvm.{dylib|so}
 - libjvm.{dylib|so} (without path)
 - Windows
 - %JAVA_HOME%\bin\server\jvm.dll
 - %JAVA_HOME%\jre\bin\server\jvm.dll
 - jvm





- New environment symbol `.dev.null` (all operating systems)
 - Meant to be used in the context of redirecting standard files for REXX commands when using `ADDRESS ... WITH`
 - Redirecting `output` or `error` to `.dev.null` will get any lines written to these standard files to be ignored
 - `input` from `.dev.null` will signal `NOTREADY` indicating no input available
 - Example for ignoring all error output to `stderr`

```
ADDRESS SYSTEM "some_command" WITH ERROR USING (.dev.null)
```





- New public routine `id2x(...)`

- Make the `identityHash` value of ooRexx objects better legible for humans if on a 64-bit system
- Will insert an underscore after eight hexadecimal character
- Example

```
id=.object~identityHash
```

```
say "id:" id id2x(id) /* maybe: id: -140601995565697 FFFF801F_8BFFE57F */
```

```
::requires BSF.CLS
```





- New public class **.BSF.Clipboard**
 - Allows access to the clipboard
 - Allows copying and pasting text, optionally supplying an encoding (e.g. “UTF-8”)
 - Allows to solve character set problems, if any, in an easy manner
 - Allows copying and pasting images (e.g. from/to the JDOR Rexx command handler)
 - E.g. creating an image with JDOR, copying it to the clipboard and later pasting it into any application (either manually or via Rexx, if the application is controlled by Rexx)
 - Allows to determine whether the clipboard is empty
 - Allows to clear the clipboard and more



Links (As of: 2024-03-03)



- ooRexx
 - <https://sourceforge.net/projects/oorex/5.1.0beta/>
- Java/OpenJDK
 - Try to get the installation package that contains *JavaFX*
 - From Amazon, Azul, IBM, Microsoft, Oracle, ...
- BSF4ooRexx850 (external ooRexx function and class package)
 - <https://sourceforge.net/projects/bsf4oorex/BSF4ooRexx-850.20240304-GA/>
 - Checkout samples that start with “1-” in BSF4ooRexx850/samples
- Slides introducing the ooRexx-Java bridge
 - <https://wi.wu.ac.at/rgf/wu/lehre/autojava/material/foils/>



BSF4ooRexx850: Roundup



- External Rexx function package
 - BSF4ooRexx version [850](#) needs at least Java [8](#) or later, and ooRexx [5.0](#) or later
 - Allows interacting with Java classes and objects
- Configuration and setup improvements, honoring the environment variables [JAVA_HOME](#) and [JDK_STARTUP_OPTIONS](#)
- **"BSF.CLS"**
 - Camouflages Java as ooRexx
 - Adds the environment symbol [.dev.null](#) (for use in [ADDRESS...WITH](#))
 - Adds the public routine [id2x\(\)](#)
 - Adds the public class [.BSF.Clipboard](#)

