The New BSF400Rexx 6.00

Rony G. Flatscher, WU
2018 International Rexx Symposium
Overview

- Brief history
  - Purpose
  - "Swiss Army Knife (SAK)" for Rexx programmers
- BSF4ooRexx version 6.0
  - New features
- Roundup and outlook
BSF4ooRexx – Brief History, 1

• Proof of concept at the University Essen 2000
  – Originally for OS/2 and Windows
  – Purpose
    • Allow OS/2 Rexx programmers to use Java as a Rexx function library to take advantage of Java e.g. for GUIs
    • Allow their Rexx programs to be run unchanged if migrated to Windows, even if they are GUI applications
  – Later for Linux and MacOSX
  – Some BSF4ooRexx GUI samples were originally created on OS/2 and still (18 years later!) run unchanged on Windows, Linux and MacOSX!
    - samples/3-070_ShootOut.rxj
BSF4ooRexx – Brief History, 2

- ooRexx 4.0
  - Introduced a new kernel and an excellently devised native API modelled after Java's JNI
  - BSF4ooRexx became able to take full advantage of ooRexx at the JNI-C++ level into both directions!
    - Allowing Rexx to interact in total new ways with Java
      - Implementing any Java interface classes in Rexx!
        - E.g. allowing Java to callback into Rexx!
      - Implementing abstract Java classes in Rexx!
      - Allowing to extend Java classes to access protected members
    - Allowing Java to interact with Rexx objects
      - Sending messages
      - Fetching even Rexx objects as return values
BSF4ooRexx – Brief History, 3

- Goal of "Swiss Army Knife (SAK)" for Rexx
  - Make good for missing external function packages
    - E.g. ssl, crc32, IPv6 long before Rexx supported it, etc.
  - Allows for creating graphical user interfaces (GUI)
    - awt, swing, and even JavaFX!
  - Any third party Java class library can be used
    - BSF4ooRexx' .Net support on Windows realized that way!

- Best of all
  - ooRexx programs can run unchanged on all operating systems
  - Fully exploiting the promise of Java "compile once, run everywhere"!
BSF4ooRexx 6.00, Java Support, 1

- Basing on Java 1.6/6.0, hence "6.00"
  - Implementation can take advantage of significant new Java features like generics
  - Support for JSR-223 (javax.script)
    - Rexx can be easily deployed by any Java application
    - Rexx can be used as a macro language wherever e.g. JavaScript, Groovy, Jython and the like gets used

- Making sure that it runs on Java 9 and later
  - Java 9 introduced some significant internal changes, breaking sometimes compatibility of reflective Java code
BSF4ooRexx 6.00, Java Support, 2

• Making sure that it runs on Java 9 and later
  – Adapting Java 9 support for MacOSX
    • Apple Java classes not accessible anymore
  – Two different reflection mechanisms, even caching!
    • java.lang.reflect based
      – Only way on Java 1.6/6, but also needed to fully use Java 1.7/7
    • java.lang.invoke based
      – MethodHandle based
      – Currently (beta phase) default for Java 1.8/8 and Java 9
    • Reflection mechanism can be switched either way at runtime
      – Performance comparable, MethodHandle slightly faster
BSF4ooRexx 6.00, Java Support, 3

- Support for ooRexx Array's `makeArray` and `supplier` semantics for Java objects that implement the Java interfaces for collections
  - `java.lang.Iterable`
  - `java.util.Collection`
  - `java.util Enumeration`
  - `java.util.Iterator`
  - `java.util.Map`
- Can therefore be directly used in `DO...OVER`!
BSF4ooRexx 6.00, External Function, 1

- **BsfCreateRexxProxy(rexx, [user], ...)**
  - Boxes Rexx object into a Java object
    - Rexx object may be
      - a plain string representing Rexx code
      - an array of strings (new)
      - a routine (new)
      - a method
    - The optional second argument is a user/programmer supplied Rexx object that gets sent back on callbacks from Java (entry "USERDATA" in the slot argument)
    - The third argument may be "R[exx]", a list of Java interfaces, the name of an abstract class followed by its arguments
BSF4ooRexx 6.00, External Function, 2

- **BsfTestPing([rep])**
  - New function to allow timing external calls
  - If `rep` (repetitions) is given, the function will call a Java testPing method `rep` times

- **New subfunc BSF("testPing" [,rep [,obj,msg] ])**
  - No argument: roundtrip from Rexx to Java
  - `rep`: Java calls repetition times native C++ function
  - `rep`, `obj`, `msg`: Java sends repetition times message `msg` to the supplied Rexx `obj`
BSF4ooRexx 6.00, FXML Enhancement

- JSR-223 invocations may not supply the file name of the program that gets run
  - Despite the documentation of `javax.script`!
  - Surprisingly `JavaFX` is one such infrastructure
    - In case of an execution error the file name of the Rexx package cannot be given, if invoked from an `FXML` file!
- Enhancement
  - The artificial Rexx file name will get the location value from the `ScriptContext` added to it
  - A Rexx programmer can therefore at least locate the source of the invocation of the Rexx program
BSF4ooRexx 6.00, BSF.CLS, 1

- Now **INTERPRET** free!
  - 18 years ago only **INTERPRET** allowed for some needed dynamic Rexx code invocation
- Using the **Routine** class allows to forgo it
- Added caching of external routines
  - Turned out to be up to 20 times faster!
  - Meanwhile ooRexx 5.0 applies even better caching and increases in lookups of environment symbols ("dot variables") including class lookups
- Once BSF4ooRexx requires ooRexx 5.0 it will therefore forgo its own caching :-)
BSF4ooRexx 6.00, BSF.CLS, 2

• New ooRexx class Slot.Argument
  - Whenever a Java callback reaches Rexx a slot argument gets added as the last argument
    • A Rexx Directory object that may contain useful entries
    • Sometimes programmers wished to be able to distinguish this slot argument from a normal Rexx Directory argument
  - Slot.Argument is a plain subclass of Directory
    • Using Object's isA(.Slot.Argument) returns .true, if the argument is indeed a slot argument!
    • Idea: Jon Wolfers at the 2017 Rexx Symposium!
BSF4ooRexx 6.00, **BSF.CLS**, 3

- MacOSX
  - ooRexx lately reports "DARWIN" as the name
    - Supplied by MacOSX
    - Before, for years "MACOSX" was supplied
  - To keep backward compatibility the entries
    - `.bsf4rexx~opSys` still will be mapped to "MACOSX"
    - `.bsf4rexx~opSys1` mapped to "M"
    - `.bsf4rexx~opSys2` mapped to "MA"
    - `.bsf4rexx~opSys3` mapped to "MAC"
BSF4ooRexx 6.00, BSF.CLS, 4

- New classes to ease GUI programming
  - FXGuiThread
    - Allows to asynchroneously send messages to GUI objects
    - Messages will be dispatched on the "JavaFX Application Thread" (the JavaFX GUI thread, see other talk)
    - Makes sure no hangs occur
  - GuiMessage
    - Modelled after ooRexx' Message class
    - Returned by FXGuiThread methods
      - Can be used to wait for the message to have been executed
      - Can be used to fetch return value, if any
BSF4ooRexx 6.00, BSF.CLS, 5

• New entries in .bsf4rexx
  - .bsf4rexx~java.version
    • The full Java version string, e.g. "1.8.0_162"
  - .bsf4rexx~java.major.version
    • "6" for Java 1.6, "7" for Java 1.7, "8" for Java 1.8, "9" for Java "9" and up, e.g. "8"
    • Eases testing for a certain Java version
  - .bsf4rexx~java.minor.version
    • Whatever the Java version string supplies as minor information, e.g. "0_162"
BSF4ooRexx 6.00, BSF.CLS, 6

- **bsf.compile(className,JavaSourceCode)**
  - Compiles supplied Java source code
  - Loads denoted `className` from the compiled Java program for further usage
  - Can be used for implementing lambda functions
    - Really only necessary, if an ooRexx implemented lambda function appears to be too slow
      - Only needed, if lambdas get employed by some Java algorithms in the ten-to-hundred-thousands-of-times
  - Can be useful for solving rare "exotic" problems
  - Support for NetRexx planned, once a comparable on-the-fly compilation becomes possible for it
BSF4ooRexx 6.00, **BSF.CLS**, 7

- New hash-bang line for all Rexx scripts
  ```bash
  #!/usr/bin/env rexx
  ```

- Unix-related
  - Allows executing Rexx scripts as Unix commands
  - One needs to set the executable bit, e.g.
    ```bash
    chmod a+x *.rex
    ```
  - `/usr/bin/env` will use the environment to find the program `rexx` to run the script
  - Hint: in order to work on Unix the lines must be terminated with \`LF ("0A"x) only!\`
Roundup and Outlook

- A lot of work on many frontiers!
- Work on BSF4ooRexx 6.0 concluded
  - All test units pass
    - Extremely important
    - Without them this work could not have been possibly be done in that time frame
- Beta test phase
  - Actually "gamma", if not already release quality
  - Planned to add on-the-fly compiling for NetRexx as mentioned in the presentation