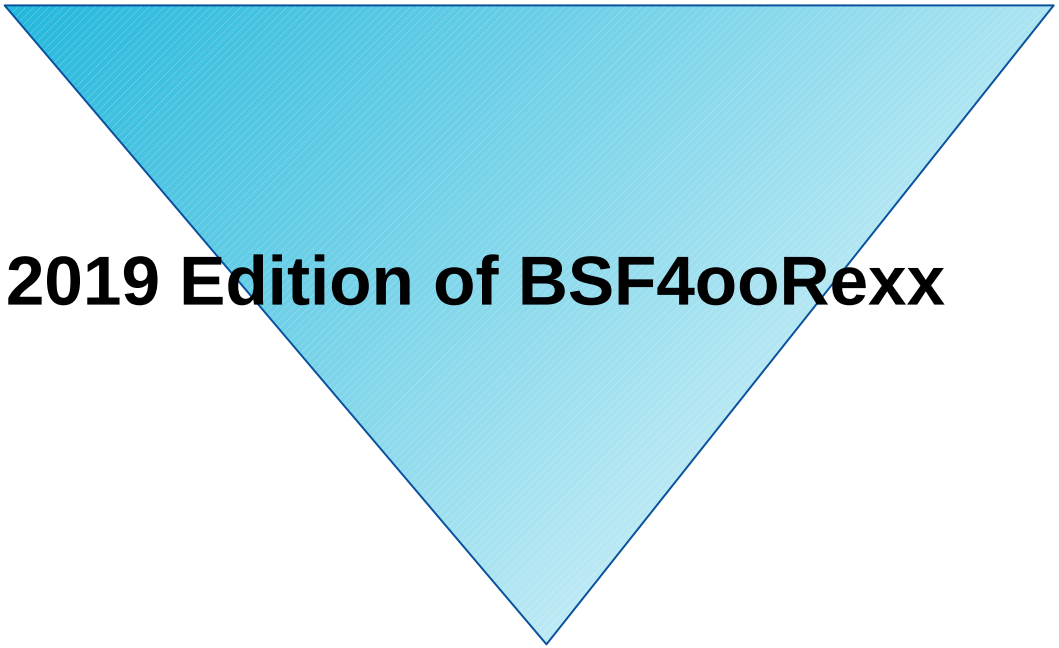


The 2019 Edition of BSF4ooRexx

2019 – International Rexx Symposium

Hursley, September 2019

Rony G. Flatscher (Rony.Flatscher@wu.ac.at, http://www.ronyRexx.net)
Wirtschaftsuniversität Wien, Austria (http://www.wu.ac.at)

Overview

- Brief history
- Bird-eye's view of BSF4ooRexx
- New Features in BSF4ooRexx 641.20190830
- Outlook
- Roundup
- Appendix: how to create a full JRE 11 from JDK 11 and OpenJavaFX 11?

Brief History, BSF4ooRexx, 1

- 2000/01: "BSF4Rexx"
  - Proof of concept at University Essen
  - Windows, OS/2
- Goals
  - Allow OS/2 Rexx programs to run on Windows, even GUI apps!
  - If possible extend to other operating systems
  - Run on all Rexx/SAA interpreters, last version:
    - 2008: <http://wi.wu.ac.at/rgf/rexx/bsf4rexx/current/>

Brief History, BSF4ooRexx, 2

- 2009: "BSF4**oo**Rexx"
  - Exploiting new native ooRexx APIs introduced with the new kernel of ooRexx 4.0
    - Hence, *not* compatible with RexxSAA anymore!
  - Among many new features, that have become possible with the native ooRexx APIs, BSF4ooRexx allows to implement abstract Java methods with ooRexx (nifty for callbacks that get handled in Rexx)!
    - Abstract Java methods from Java interface classes can be fully implemented with ooRexx methods!
    - Java abstract classes can be extended by ooRexx classes!

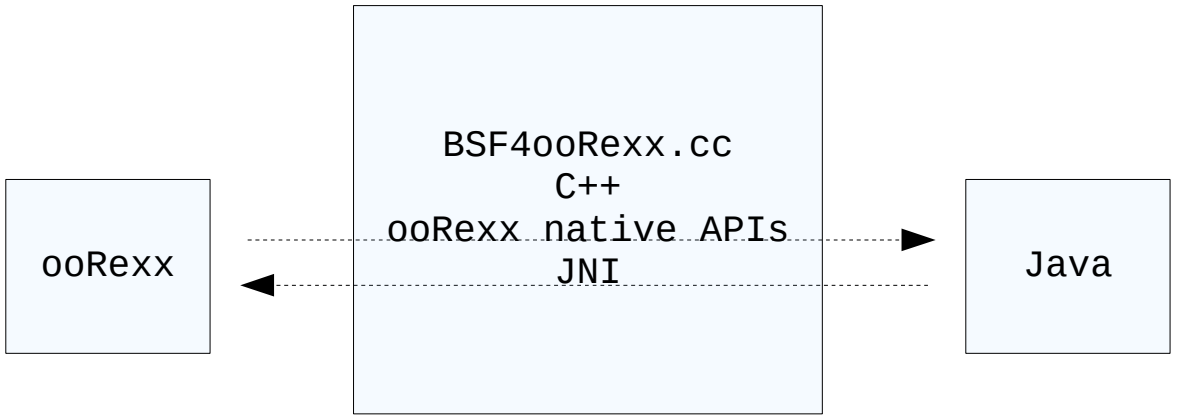
Brief History, BSF4ooRexx, 3

- 2019: "BSF4**oo**Rexx", some base information
  - Baseline for ooRexx: version **4.1**
  - Baseline for Java: version **6** (used to be the outdated version 1.4)
  - Hence BSF4ooRexx version "**641**.YYYYMMDD"
  - BSF4ooRexx kernel rewritten
    - Using java.lang.reflect *and* (new) java.lang.invoke
    - Allows Java 6, 7, 8 and new *modular* Java 9, 10, 11, ...
  - Planned to be released in 2019

Bird-Eye's view of BSF4ooRexx, 1

- External ooRexx function package
  - C++ (ooRexx native APIs)
    - Maintains ooRexx objects for Java interaction in a registry
  - JNI (C++ bridge to Java)
  - Java ooRexx support package (set of Java classes)
    - Maintains Java objects for ooRexx interaction in a registry
  - ooRexx support package (program) BSF.CLS
    - Turns Java into a dynamic, message based language
    - Makes it easy for Rexx programmers to exploit Java

Bird-Eye's view of BSF4ooRexx, 2



Bird-Eye's view of BSF4ooRexx, 3 Example, 1

- Use the Java class java.awt.Dimension
  - Find documentation about that Java class on the net
    - All Java classes get documented in interlinked HTML pages!
  - Javadoc on the Internet, search for e.g.,

    ```
    java 11 javadoc java.awt.Dimension
    ```

  - Could yield a link like
    `https://docs.oracle.com/en/java/javase/11/docs/api/java.desktop/java/awt/Dimension.html`
  - Stores the width and height dimension
  - Has methods to change the field (attribute) values

Bird-Eye's view of BSF4ooRexx, 4 Example, 2

- An example

```
dim=.bsf~new("java.awt.Dimension", 123,456)  -- create Dimension object
say "width: " dim~width                      -- get width value
say "height:" dim~height                     -- get height value
say "dim~toString:" pp(dim~toString)         -- use Java method toString()
say
say "changing dimension to 777x888 ..."
dim~setSize(777,888)                         -- use Java method setSize(...)
say "width: " dim~width                      -- get width value
say "height:" dim~height                     -- get height value
say "dim~toString:" pp(dim~toString)         -- use Java method toString()

::requires "BSF.CLS" -- get the Java bridge, camouflage Java as ooRexx
```

- Output

```
width:  123
height: 456
dim~toString: [java.awt.Dimension[width=123,height=456]]

changing dimension to 777x888 …
width:  777
height: 888
dim~toString: [java.awt.Dimension[width=777,height=888]]
```

New Features in BSF4ooRexx, 1

- Most important changes and additions since the last International Rexx Symposium
  - For a complete list please consult the log entries in the files (located in the BSF4ooRexx install directory)
    - changesBSF4ooRexx.txt
    - changesOOo.txt
- Demonstration of some new features
  - With nutshell examples or pointing to sample programs installed with BSF4ooRexx
    - Hint: Load the file samples/index.html into your browser!

New Features in BSF4ooRexx, 2
Character Set Translations, 1

- New features in BSF.CLS
  - New public routine bsf.iconv(str,fromCp,toCp)
    - Allows character set translations of the string (str) from a given encoding (fromCP) to another encoding (toCp), returns supplied string (str) in the other encoding (toCp)
    - E.g. translating an 8-Bit Windows encoded string to UTF-16 encoding or vice-versa
    - Examples
      - List charsets available in your current JRE

        `samples\1-040_list_charsets.rxj`
      - Demonstrate usage of bsf.iconv(...)

        `samples\1-080_charsetTranslations.rxj`

New Features in BSF4ooRexx, 3
Character Set Translations, 2

- An example

```
str="hi," || "0d0a"x || "there!"     -- note CR-LF embedded
say "str:          " pp(str)         -- show string
say "c2x(str):     " pp(c2x(str))    -- show string in hex
strUtf16=bsf.iconv(str, "cp850", "utf-16")   -- change to UTF-16
say "strUtf16:     " pp(strUtf16)    -- show UTF-16 string
say "strUtf16~c2x:" pp(strUtf16~c2x)-- show UTF-16 string in hex

::requires "BSF.CLS" -- get the Java bridge, camouflage Java as ooRexx
```

- Output (Windows, code page 850)

```
str:          [hi,
There!]
c2x(str):     [6869 2C 0D 0A 74 68 65 72 65 21]
strUtf16:     [■  h i ,
 t h e r e !]
strUtf16~c2x: [FEFF 0068 0069 002C 000D 000A 0074 0068 0065 0072 0065 0021]
```

BOM
(byte order mark)

New Features in BSF4ooRexx, 4
New Class "AwtGuiThread"

- New features in BSF.CLS
  - New public class AwtGuiThread
    - Class that allows to update awt/swing GUIs asynchroneously
    - Exactly the same protocol as in the public class FxGuiThread
      - Cf. <http://www.rexxla.org/events/2018/schedule.html>
        - Slides+article: "The New BSF4ooRexx 6.00" and
        - Slides+article: "Anatomy of a GUI (Graphical User Interface) Application for Rexx Programmers"
    - Example
      - Demonstrate usage of AwtGuiThread

        `samples\3-090_update_awtSwing_GUI-from-non-GUI-thread.rxj`
      - Example comparable to

        `samples\javafx\javafx_update_GUI-from-non-GUI-thread.rxj`

New Features in BSF4ooRexx, 5
New Entries in .BSF4Rexx Directory

- New entries relating to BSF.CLS
  - ".bsf4rexx~**boolean.true**", ".bsf4rexx~**boolean.false**"
    - Returns the cached java.lang.Boolean truth values
  - ".bsf4rexx~**bsf.cls.fullPath**"
    - Returns the fully qualified path to BSF.CLS in use
  - ".bsf4rexx~**bsf.cls.location**"
    - Returns the location of BSF.CLS in use
  - ".bsf4rexx~**display.version**"
    - Returns a string with version information, e.g.,

      "ooRexx 5.0.0 r11909 (30 Aug 2019) / BSF 641.20190830 / Java 1.8.0_171, 32-bit"
  - ".bsf4rexx~**rexx.version**"
    - Returns the current Rexx version as a decimal number in the form "major.minor", e.g. "4.1", "4.2", "5.0"

New Features in BSF4ooRexx, 6
Context ClassLoader Related

- New features in BSF.CLS
  - New public routine bsf.contextClassLoader([urlOrDirOrFileName | urlCollection])
    - Without arguments
      - returns current Java thread context ClassLoader
    - With arguments
      - New Java URLClassLoader gets created that also looks up the supplied locations in addition
      - The current Java thread context ClassLoader gets set to the new one, which then gets returned by the routine
    - Examples
      - Sample that employs the new routine

        `samples\JavaFX\fxml_05\staff.rxj`

New Features in BSF4ooRexx, 7

Box.StrictArg()

- New features in BSF.CLS
  - Added, just in case future class/type musings in future Java versions need this :-)
  - New public routine box.strictArg(type,value[,bPrimitive])
    - Allows to pick strictly by the supplied type using an instance of the new Java class RexxStrictArgument
      - type either a specific Java class, but may also be one of the indicator strings from box()
    - value to be boxed
    - bPrimitive: if .true and value is primitive, then type of candidate method/constructor arguments must be primitive

New Features in BSF4ooRexx, 8
Changed bsf.import()

- New features in BSF.CLS
  - Changed public routine bsf.import(javaClass)
    - Will *not* import abstract Java classes anymore!
      - Will raise a condition with the advice to use the routine bsf.loadClass(javaClass) instead
      - Reasoning
        - bsf.import() will add the ooRexx new class method, although abstract Java classes cannot be instantiated as the presence of an ooRexx new class method implies
    - Use bsf.loadClass(javaClass) instead

New Features in BSF4ooRexx, 9

Changed RexxScriptEngine

- Changed org.rexxla.bsf.engines.rexx.jsr223.RexxScriptEngineFactory
  - Defined additional mime types: "text/rexx", "text/oorexx", "application/rexx", "application/oorexx"
  - Running "samples\Java\javax.script\RexxRunScript.rex -i" therefore yields about the ooRexx Java script engine e.g.,

```
ooRexx
        getEngineName           : Open Object Rexx (ooRexx)
        getEngineVersion        : 100.20190726
        getExtensions           : [rex, rexx, orx, cls, rxj, rexxj, jrexx, rxo]
        getLanguageName         : ooRexx
        getLanguageVersion      : REXX-ooRexx_5.0.0(MT)_32-bit 6.05 30 Aug 2019
        getMimeTypes            : [text/rexx, text/oorexx, application/rexx,
application/oorexx, text/x-rexx, text/x-rexx-java, text/x-rexx-java-ooo]
        getNames                : [rexx, Rexx, oorexx, ooRexx, orexx, oRexx]
        getParameter(THREADING): MULTITHREADED
```

Outlook

- Release planned for 2019
  - Last version that supports ooRexx 4.1.x and 4.2.x
  - Next version of BSF4ooRexx will be based on ooRexx 5.0!
    - Adding e.g., redirectable command handlers at runtime that can be implemented either in Java, NetRexx or BSF4ooRexx
- Possible changes
  - Changing the package name of the ooRexx Apache OpenOffice (AOO)/LibreOffice (LO) support to the org.rexxla namespace
  - Changing the installation/uninstallation logic for AOO/LO

Roundup

- New Features in BSF4ooRexx 641.20190830
  - Since 2014 in the works
    - Based on ooRexx 4.1 and Java 6
      - Some JavaFX samples need ooRexx 5 for stability reasons
    - BSF4ooRexx kernel reworked to support modular Java
    - Plentiful of new features, utility classes and utility routines
    - Still easy to learn and to use
    - Installation package can be directly used for
      - Windows (32/64bit), Linux (32/64bit), MacOS (32/64bit), *IBM s390x* :)
  - Questions?

    … hang on ! ;)

URLs

- RexxLA-Homepage (non-profit SIG, owner of ooRexx, BSF4ooRexx)

  <http://www.rexxla.org/>
- ooRexx 5.0 beta on Sourceforge

  <https://sourceforge.net/projects/oorexx/files/oorexx/5.0.0beta/>
- BSF4ooRexx on Sourceforge (ooRexx-Java bridge)

  <https://sourceforge.net/projects/bsf4oorexx/>
- Introduction to ooRexx (254 pages)

  <https://www.facultas.at/Flatscher>
- JetBrains "IntelliJ IDEA", powerful IDE for all operating systems
  - <https://www.jetbrains.com/idea/download>, free "Community-Edition"
  - Alexander Seik's ooRexx-Plugin with readme (as of: 2019-08-27)
    - <https://sourceforge.net/projects/bsf4oorexx/files/Sandbox/aseik/ooRexxIDEA/beta/1.0.5/>

Appendix: Howto Create JRE 11 with JavaFX11, 1

- Java modules introduced with Java 9
  - Only distributed as "JDK" Java development kit
    - Environment needs to be adjusted to modules
      - Needs may be different at compile and runtime!
      - Idea: create smallest possible footprint for Java applications by using only the needed modules!
  - JavaFX donated to the opensource community by Oracle
  - JDK11 *removed* JavaFX!
    - Download opensource JavaFX modules from Gluon
      - https://gluonhq.com/products/javafx/

Appendix: Howto Create JRE 11 with JavaFX11, 2

- Need for a full Java runtime environment (JRE), e.g.,
  - Server configurations where many different servlets need many different Java modules
  - Scripting, ad-hoc programs
    - Unforeseeable need for Java modules
- JDK comes with a tool named jlink
  - Allows to create a tailored Java runtime environment
  - Can be used to create a full JRE from any modular JDK!

Appendix: Howto Create JRE 11 with JavaFX11, 3

- Steps
  - Download JDK11 (e.g. https://adoptOpenJDK.net)
    - Locate JDK home directory and assign it to the JAVA_HOME environment variable
    - All JDK modules in: $JAVA_HOME/jmods
  - Download Open JavaFX
    - Locate JavaFX directory and assign it to the environment FX_DIR variable
    - All JavaFX modules in: $FX_DIR/jmods

Appendix: Howto Create JRE 11 with JavaFX11, 4

- Steps
  - Open a command line/terminal window
    - Define environment variables

```
set JAVA_HOME=path-to-JDK-home
set FX_DIR=path-to-JavaFX-directory
```

    - Issue the jlink command (Windows)

```
%JAVA_HOME%\bin\jlink -p %JAVA_HOME%\jmods,%FX_DIR%\jmods --add-modules ALL-MODULE-PATH --output tgtdir
```

    - Issue the jlink command (Unix)

```
$JAVA_HOME/bin/jlink -p $JAVA_HOME/jmods,$FX_DIR/jmods --add-modules ALL-MODULE-PATH --output tgtdir
```

- 'tgtdir' will contain the appropriate JRE with all modules from JDK and from JavaFX!

```
Windows: tgtdir\bin\java --list-modules
Unix:    tgtdir/bin/java --list-modules
```