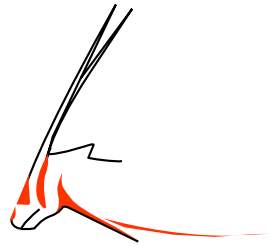
BSF4ooRexx 6.41 Going GA



2021 – International Rexx Symposium

Online ("Covid-19")

November 7th – November 11th 2021

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

Wirtschaftsuniversität Wien, Austria (http://www.wu.ac.at)

Overview

- Brief history
- Bird-eyes view of BSF4ooRexx
  - Nutshell examples to demonstrate utility
- Some significant new features
- Planned release schedule
- Roundup and outlook
- Questions

Brief History, BSF4ooRexx, 1

- 2000: proof-of-concept at University of Essen
  - Bridging Rexx and Java with Apache BSF
  - Platform independent: OS/2, Windows, Unix
  - Using the Rexx SAA APIs, hence Regina, ooRexx, ...
- 2009: switching to ooRexx, renamed to BSF4***oo***Rexx
  - ooRexx 4.0 introduced powerful new C++ APIs
  - Implementing callbacks from Java to Rexx became possible
- End of 2021: BSF4ooRexx in Gamma, about to go GA

Brief History, BSF4ooRexx, 2

- 2012: release 410.20120618
  - Information encoded in version string

```
parse value 410.20120618 with version "." date
```

    - Version/100: 4.10 (minimum version of ooRexx)
- 2013: release 410.20130107
- 2013: release 411.20130714
- 2014: release 450.20140914
  - `Note: this version still runs with ooRexx 4.1`
- No official GA release since then (waiting on ooRexx 5.0 to go GA)

Brief History, BSF4ooRexx, 3

- BSF4ooRexx comes with support for
  - OpenOffice.org (OOo), all platforms
    - OOo is now Apache OpenOffice (AOO)
    - LibreOffice (LO), a fork of OOo
  - ooRexx package UNO.CLS
    - Eases programming considerably
  - ooRexx can be even used as a macro language for AOO and LO
  - AOO/LO allow using MS Office files (e.g. Word, Excel)
  - Cf. *bsf4oorexx/samples/OOo*

Brief History, BSF4ooRexx, 4

- BSF4ooRexx comes with support for
  - Microsoft's *.Net* on Windows
    - ooRexx package CLR.CLS
      - "common language runtime"
      - Eases programming considerably
      - Modelled after BSF.CLS therefore effectively camouflaging *.Net* as ooRexx
    - Cf. *bsf4oorexx/samples/clr*

Bird-Eyes View of BSF4ooRexx

- Rexx
  - Makes it possible to access and exploit all of Java via an external Rexx function package
  - Creating Java objects, invoking methods
- ooRexx
  - Camouflages Java as ooRexx using BSF.CLS
  - Allows for implementing Java methods in ooRexx
- Java
  - Makes it possible to run Rexx programs
  - Allows for sending Rexx objects messages

Rexx External Function Package
Very Brief Overview

- Classic Rexx interface style
- Exploiting all of Java becomes possible
- Responsibility of releasing Java objects
  - Java objects are maintained in a Java directory
  - Once Rexx does not need a Java object anymore, it needs to get unregistered by Rexx

Rexx External Function Package

'java.awt.Dimension' Example

- Java class (type) named *java.awt.Dimension*
  - Allows for maintaining a *width* and *height* value
  - Documentation on Java classes can be easily found by searching the Internet using the keyword "javadoc" (a Java tool) and the class name, e.g.

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

    → `https://docs.oracle.com/javase/8/docs/api/java/awt/Dimension.html`
- Rexx will create an instance (a value, an object) and supplies two arguments for the *width* and *height*
- Rexx then uses the Java *toString* method

Rexx External Function Package

'java.awt.Dimension' Example

- Example

```
if rxFuncQuery("BSF") = 1 then   /* BSF() support not loaded yet ? */
do
   call rxFuncAdd "BsfLoadFuncs", "BSF4ooRexx", "BsfLoadFuncs"
   call BsfLoadFuncs
   call BsfLoadJava
end

say "creating a Java value (object) of type 'java.awt.Dimension' ..."
dim=bsf('new', , 'java.awt.Dimension', 300, 400)

say "invoking Java 'toString' method: " bsf("invoke", dim, 'toString')

call bsf "unregisterBean", dim   /* make sure to unregister Java object */
```

- Output

```
creating a Java value (object) of type 'java.awt.Dimension' ...
invoking Java 'toString' method:  java.awt.Dimension[width=300,height=400]
```

Camouflaging Java as ooRexx Package BSF.CLS

- Java objects get represented with ooRexx proxy objects
  - Instances of the ooRexx class *BSF*
- Java objects get automatically released, if the ooRexx proxy object is not used anymore
- The ooRexx message based interface style becomes available for interacting with Java objects

A Little Remark About ooRexx

Messages to Interact with Objects, 1

- ooRexx uses the message metaphor (cf. Smalltalk)
  - *Everything* is an object
    - Therefore a plain string is an object as well
  - An object is like a living thing such that one is able to interact with it by *sending* it *messages* (**~**)
    - The *object will look for a method* by the name of the received message *and invokes it*
    - Any return value will be returned *by the object*
  - "*object*" is just a synonym for "*value*", "*instance*" !

A Little Remark About ooRexx

Messages to Interact with Objects, 2

- An example

```
test="12345"          /* a string */
say "REXX-style (function invocation): say reverse(test)"
say reverse(test)     /* invoking the built-in string function "reverse" */
say

say "ooRexx-style (sending a message): say test~reverse"
say test~reverse      /* invoking the string method "reverse"           */
```

- Output

```
REXX-style (function invocation): say reverse(test)
54321

ooRexx-style (sending a message): say test~reverse
54321
```

Using BSF.CLS

'java.awt.Dimension' Example

- Example

```
say "creating a Java value (object) of type 'java.awt.Dimension' ..."
dim=.bsf~new('java.awt.Dimension', 300, 400)

say "invoking Java 'toString' method: " dim~toString

/* ooRexx will automatically unregister any Java object! */

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

- Output

```
creating a Java value (object) of type 'java.awt.Dimension' ...
invoking Java 'toString' method:  java.awt.Dimension[width=300,height=400]
```

Using BSF.CLS

Some Remarks

- The camouflaging support also makes it easy to
  - Create (strictly typed) Java arrays
    - Interact with Java arrays as if they were ooRexx arrays
    - Allows applying ooRexx do...over
    - Adds ooRexx makearray and supplier methods
  - Use platform independent GUI popups
  - ... and much more ...

ooRexx

Example: Two Dimensional ooRexx Array

- Example

```
-- create a two-dimensional (initially 5x10) Rexx array
arr=.array~new(5,10)

arr[1,1]="First Element in Rexx array."       -- put an element
arr~put("Last Element in Rexx array.", 5, 10) -- put another one

do element over arr  -- loop over elements in array
   say element
end
```

- Output

```
First Element in Rexx array.
Last Element in Rexx array.
```

BSF4ooRexx

Example: Two Dimensional *Java* Array

- Example

```
-- create a two-dimensional (5x10) Java Array of type String
arr=.bsf~bsf.createJavaArray("java.lang.String", 5, 10)

arr[1,1]="First Element in Java array."       -- put an element
arr~put("Last Element in Java array.", 5, 10) -- put another one

do element over arr  -- loop over elements in array
   say element
end

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

- Output

```
First Element in Java array.
Last Element in Java array.
```

Using BSF.CLS

Dialog Window Class "BSF.DIALOG"

- ooRexx class using the portable Java GUI system
  - Runs on all operating systems unchanged
    - Windows, Apple, Linux, ...
  - Easy to use for ooRexx programmers
  - Supplies three methods
    - messageBox() to inform the user
    - dialogBox() to get information from the user
    - inputBox() to get information from the user
  - Cf. BSF4ooRexx supplied sample named
    - bsf4oorexx/samples/1-020_demo.BSF.dialog.rxj

Using BSF.CLS

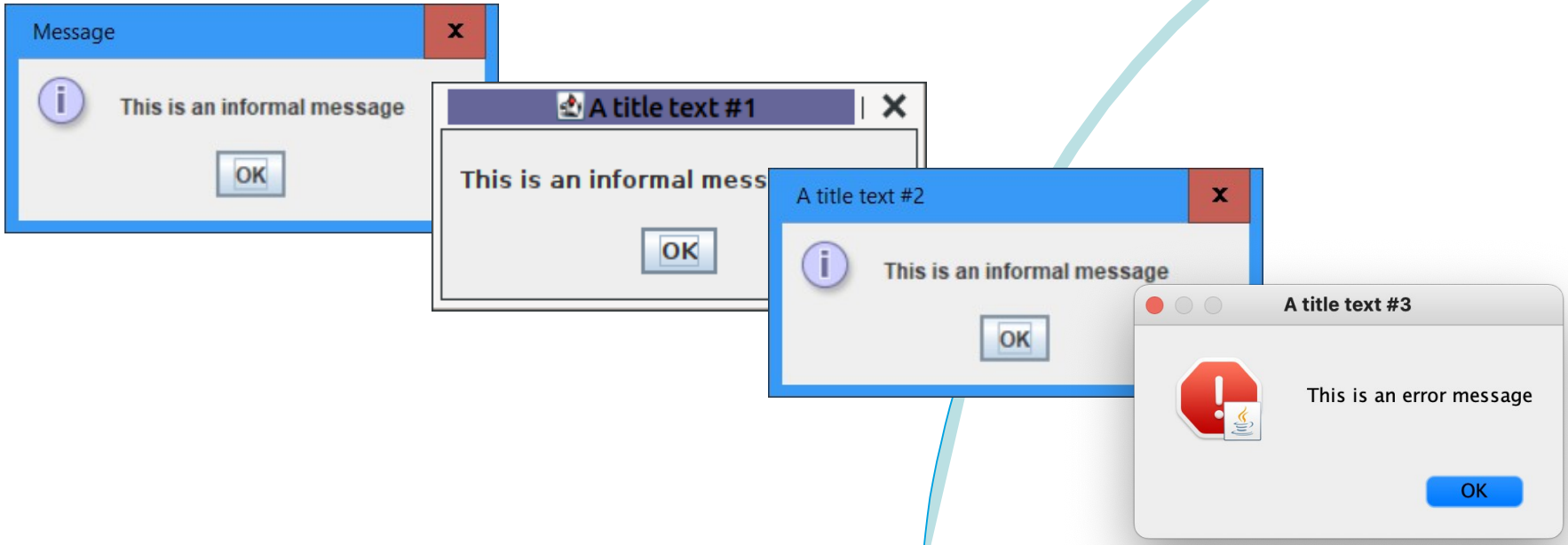
Example: Using messageBox() from BSF.DIALOG

- Example

```
-- cf. bsf4oorexx/samples/1-020_demo.BSF.dialog.rxj
say "Demonstrating .bsf.dialog~messageBox(...):"

/* args: message, title, messageType */
.bsf.dialog~messageBox("This is an informal message")
.bsf.dialog~messageBox("This is an informal message", "A title text #1")
.bsf.dialog~messageBox("This is an informal message", "A title text #2", "info")
.bsf.dialog~messageBox("This is an error message",    "A title text #3", "error")

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



Using BSF.CLS

Example: Using dialogBox() from BSF.DIALOG

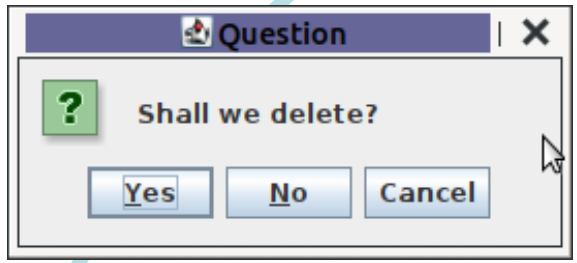
- Example

```
-- cf. bsf4oorexx/samples/1-020_demo.BSF.dialog.rxj
say "Demonstrating .bsf.dialog~dialogBox(...):"

/* args: message, title, messageType, optionType, icon, buttons, default */
res=.bsf.dialog~dialogBox("Shall we delete?", , "question",  "YesNoCancel")
say "dialogBox: you picked button #" res

txtButtons=.list~of("Tickle Alice", "Tickle Berta", "Tickle Cindy")
defButton ="Tickle Berta"
res=.bsf.dialog~dialogBox("Please pick a button", , "question", , , txtButtons, defButton)
say "dialogBox: you picked button #" res

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



- Output

```
Demonstrating .bsf.dialog~dialogBox(...):
dialogBox: you picked button # 1
dialogBox: you picked button # 2
```

Using BSF.CLS

Example: Using inputBox() from BSF.DIALOG

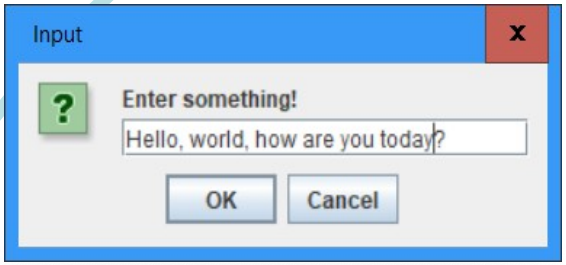
- Example

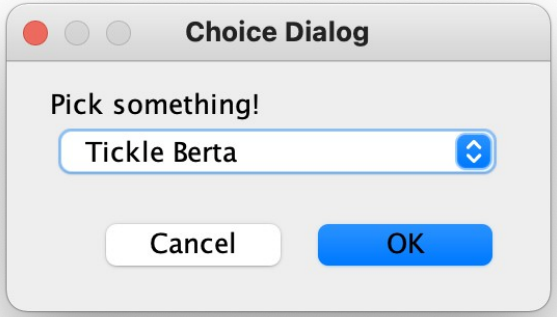
```
-- cf. bsf4oorexx/samples/1-020_demo.BSF.dialog.rxj
say "Demonstrating .bsf.dialog~inputBox(...):"

/* args: message, title, messageType, icon, textOfOptions, defaultValue */
res=.bsf.dialog~inputBox("Enter something!")
say "inputBox: you entered" pp(res)

choices=.list~of("Tickle Alice", "Tickle Berta", "Tickle Cindy")
default="Tickle Berta"
res=.bsf.dialog~inputBox("Pick something!", "Choice Dialog", "plain", , choices, default)
say "inputBox: you picked" pp(res)

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





- Output

```
Demonstrating .bsf.dialog~inputBox(...):
inputBox: you entered [Hello, world, how are you today?]
inputBox: you picked [Tickle Berta]
```

Java Employing Rexx

- Java can run Rexx programs using
  - Apache BSF (Bean Scripting Framework)
    - Open source Java scripting framework
    - Developed originally by IBM at the end of the 90's
      - Donated to the Apache Software Foundation (ASF)
  - Java scripting framework ("JSR-223")
    - Developed for the Java specification request # 223
      - Presenter served as expert
    - Introduced 2006 with Java 6
      - Package javax.script

Java Employing Rexx

Example: *Apache BSF* (Original), 1

- Java program loads a Rexx script engine using BSF
  - Requires BSF.CLS to load the camouflaging support
  - Defines ooRexx code inline
    - Will fetch the argument, sort the collection and dump the collection index and its collection value
  - Queries the Java system properties
  - Runs the ooRexx program and supplies it the Java system properties as an argument
  - Exits the Java program

Java Employing Rexx

Example: *Apache BSF* (Original), 2

```
import org.apache.bsf.*;    // Apache Bean Scripting Framework
import java.util.Vector;    // needed for arguments to script

class Code13_java_bsf
{
    public static void main (String args[]) throws BSFException
    {
       BSFManager mgr    = new BSFManager();
       BSFEngine  engine = mgr.loadScriptingEngine("rexx");
       // BSF4ooRexx: establish the ooRexx-Java bridge via ooRexx
       engine.apply ("code01", 0, 0, "::requires BSF.CLS", null, null);

       // ooRexx code to interact with the Java object argument
       String code= "use arg javaColl                              \n" +
              "tm=.bsf~new('java.util.TreeMap') /* ordered   */ \n" +
              "tm~putAll(javaColl)     /* add all properties */ \n" +
              "do counter i with index p item val over tm         \n" +
              "   say i'.' pp(p)':' pp(val)                       \n" +
              "end                                                \n" +
              "::requires 'BSF.CLS' /* get ooRexx-Java bridge */\n" ;
       // set argument for script
       Vector codeArgs = new Vector();
       codeArgs.addElement( System.getProperties() );
       engine.apply ("code02", 0, 0, code, null, codeArgs);
       mgr.terminate();
       System.exit(0);
    }
}
```

Java Employing Rexx

Example: *Apache BSF* (Original), 3

```
1. [awt.toolkit]: [sun.awt.windows.WToolkit]
2. [file.encoding]: [Cp1252]
3. [file.encoding.pkg]: [sun.io]
4. [file.separator]: [\]
5. [java.awt.graphicsenv]: [sun.awt.Win32GraphicsEnvironment]
6. [java.awt.printerjob]: [sun.awt.windows.WPrinterJob]
7. [java.class.path]: [C:\Program Files (x86)\BSF4ooRexx\bsf4ooRexx-v641-20211004-bin.jar;C:\Program Files
(x86)\BSF4ooRexx\jni4net.j-0.8.8.0.jar;C:\Program Files (x86)\BSF4ooRexx\oorexx.net.jar;.;C:\Program Files
(x86)\OpenOffice 4\program\;C:\Program Files (x86)\OpenOffice 4\program\classes\unoil.jar;C:\Program Files
(x86)\OpenOffice 4\program\classes\ridl.jar;C:\Program Files (x86)\OpenOffice
4\program\classes\jurt.jar;C:\Program Files (x86)\OpenOffice 4\program\classes\juh.jar]
8. [java.class.version]: [52.0]
9. [java.endorsed.dirs]: [E:\jre\jre008\jre1.8.0_161_32\lib\endorsed]
10. [java.ext.dirs]: [E:\jre\jre008\jre1.8.0_161_32\lib\ext;C:\WINDOWS\Sun\Java\lib\ext]
11. [java.home]: [E:\jre\jre008\jre1.8.0_161_32]
12. [java.io.tmpdir]: [C:\Users\ADMINI~1\AppData\Local\Temp\]
13. [java.library.path]: [E:\jre\jre008\jre1.8.0_161_32\bin;C:\WINDOWS\Sun\Java\bin; … cut ...
14. [java.runtime.name]: [Java(TM) SE Runtime Environment]
15. [java.runtime.version]: [1.8.0_161-b12]
16. [java.specification.name]: [Java Platform API Specification]
17. [java.specification.vendor]: [Oracle Corporation]
18. [java.specification.version]: [1.8]
19. [java.vendor]: [Oracle Corporation]
20. [java.vendor.url]: [http://java.oracle.com/]
21. [java.vendor.url.bug]: [http://bugreport.sun.com/bugreport/]
22. [java.version]: [1.8.0_161]
23. [java.vm.info]: [mixed mode]
… cut …
```

Java Employing Rexx

Example: javax.script/JSR-223 (Standard)

- Java program loads a Rexx script engine using JSR-223
  - Defines ooRexx code inline
    - Will fetch the argument, sort the collection and dump the collection index and its collection value
  - Queries the Java system properties
  - Runs the ooRexx program and supplies it the Java system properties as an argument
  - Exits the Java program

Java Employing Rexx

Example: javax.script/JSR-223 (Standard), 2

```
import javax.script.*;      // JSR-223

class Code14_java_jsr223
{
    public static void main (String args[]) throws ScriptException
    {
        ScriptEngineManager mgr = new ScriptEngineManager();
        ScriptEngine     engine = mgr.getEngineByName("rexx");
        ScriptContext        sc = engine.getContext();

        // ooRexx code to interact with the Java object argument
        String code= "use arg javaColl                              \n" +
               "tm=.bsf~new('java.util.TreeMap') /* ordered   */ \n" +
               "tm~putAll(javaColl)     /* add all properties */ \n" +
               "do counter i with index p item val over tm         \n" +
               "   say i'.' pp(p)':' pp(val)                       \n" +
               "end                                                \n" +
               "::requires 'BSF.CLS' /* get ooRexx-Java bridge */\n" ;
        // set argument for script (an Array of Object)
        sc.setAttribute(ScriptEngine.ARGV,
                        new Object[] { System.getProperties() },
                        ScriptContext.ENGINE_SCOPE);
        engine.eval(code);
        System.exit(0);
    }
}
```

Java Employing Rexx

Example: javax.script/JSR-223 (Standard), 3

```
REXXout>1. [awt.toolkit]: [sun.awt.windows.WToolkit]
REXXout>2. [file.encoding]: [Cp1252]
REXXout>3. [file.encoding.pkg]: [sun.io]
REXXout>4. [file.separator]: [\]
REXXout>5. [java.awt.graphicsenv]: [sun.awt.Win32GraphicsEnvironment]
REXXout>6. [java.awt.printerjob]: [sun.awt.windows.WPrinterJob]
REXXout>7. [java.class.path]: [C:\Program Files (x86)\BSF4ooRexx\bsf4ooRexx-v641-20211004-bin.jar;C:\Program
Files (x86)\BSF4ooRexx\jni4net.j-0.8.8.0.jar;C:\Program Files (x86)\BSF4ooRexx\oorexx.net.jar;.;C:\Program
Files (x86)\OpenOffice 4\program\;C:\Program Files (x86)\OpenOffice 4\program\classes\unoil.jar;C:\Program
Files (x86)\OpenOffice 4\program\classes\ridl.jar;C:\Program Files (x86)\OpenOffice
4\program\classes\jurt.jar;C:\Program Files (x86)\OpenOffice 4\program\classes\juh.jar]
REXXout>8. [java.class.version]: [52.0]
REXXout>9. [java.endorsed.dirs]: [E:\jre\jre008\jre1.8.0_161_32\lib\endorsed]
REXXout>10. [java.ext.dirs]: [E:\jre\jre008\jre1.8.0_161_32\lib\ext;C:\WINDOWS\Sun\Java\lib\ext]
REXXout>11. [java.home]: [E:\jre\jre008\jre1.8.0_161_32]
REXXout>12. [java.io.tmpdir]: [C:\Users\ADMINI~1\AppData\Local\Temp\]
REXXout>13. [java.library.path]: [E:\jre\jre008\jre1.8.0_161_32\bin;C:\WINDOWS\Sun\Java\bin; … cut ...
REXXout>14. [java.runtime.name]: [Java(TM) SE Runtime Environment]
REXXout>15. [java.runtime.version]: [1.8.0_161-b12]
REXXout>16. [java.specification.name]: [Java Platform API Specification]
REXXout>17. [java.specification.vendor]: [Oracle Corporation]
REXXout>18. [java.specification.version]: [1.8]
REXXout>19. [java.vendor]: [Oracle Corporation]
REXXout>20. [java.vendor.url]: [http://java.oracle.com/]
REXXout>21. [java.vendor.url.bug]: [http://bugreport.sun.com/bugreport/]
REXXout>22. [java.version]: [1.8.0_161]
REXXout>23. [java.vm.info]: [mixed mode]
… cut …
```

Java and ooRexx Objects

Example: Java Sending ooRexx Messages, 1

- Demonstrates interaction with ooRexx objects
  - Java program loads a Rexx script engine using JSR-223
  - Runs an ooRexx program, which creates and returns an instance of the ooRexx *DateTime* class
  - Java fetches the ooRexx object and sends it messages to run methods on behalf of Java
  - Java displays string values to become able to study the effects of the messages
  - Java can do *anything* with ooRexx objects! :-)

Java and ooRexx Objects

Example: Java Sending ooRexx Messages, 2

```
import javax.script.*;                // JSR-223
import org.apache.bsf.BSFException;
import org.rexxla.bsf.engines.rexx.RexxProxy;

class Code15_java_datetime
{
    public static void main (String args[]) throws ScriptException, BSFException
    {
        ScriptEngineManager mgr = new ScriptEngineManager();
        ScriptEngine     engine = mgr.getEngineByName("rexx");
        RexxProxy dt = (RexxProxy) engine.eval("return .dateTime~new");
        String strOffset = string((RexxProxy) dt.sendMessage0("offset"));
        String strDt = string(dt);
        Object strUtcIso = dt.sendMessage0("utcIsoDate");
        String strUtcTime = string((RexxProxy) dt.sendMessage0("toUtcTime"));
        System.out.println("dt\t  =["+dt+"] \noffset    =["+strOffset+"]");
        System.out.println("dateTime  =["+strDt+"] \nutcIsoDate=["+strUtcIso+"]");
        System.out.println("toUtcTime =["+strUtcTime+"] (UTC time)");
        System.exit(0);
    }
        // invoke the Rexx object's string method, return the resulting string
    static String string (RexxProxy rp) throws BSFException
    {   // send the Rexx object the message "string"
        return (String) rp.sendMessage0("string");
    }
}
```

```
dt        =[org.rexxla.bsf.engines.rexx.RexxProxy@1c29bfd]
offset    =[01:00:00.000000]
dateTime  =[2021-11-07T13:26:59.183000]
utcIsoDate=[2021-11-07T13:26:59.183000+01:00]
toUtcTime =[2021-11-07T12:26:59.183000] (UTC time)
```

BSF4ooRexx 6.41 Going GA

Some Important Changes, 1

- Information encoded in version string, changed

```
parse value 641.YYYYMMDD with version "." date
```

  - Minimum Java version: `version%100 = 6` (2006)
  - Minimum ooRexx version: `version//100/10 = 4.1` (2011)
- BSF4ooRexx reflection core for Java
  - From Java 1 (end of 90's) on using *java.lang.reflect*
  - Incompatible changes with Java 9 ("modular Java")
    - Need for adding a totally new reflection core!
    - From scratch, using *java.lang.invoke* from Java 8 on

BSF4ooRexx 6.41 Going GA

Some Important Changes, 2

- Fetching Java string values as Java objects
  - New methods with JSO (Java string object) in name
  - Allows for invoking all Java String methods
- Character set translations
  - New routine *bsf.iconv(string,fromCP,toCP)*
  - cf. *bsf4oorexx/samples/1-080_charsetTranslations.rxj*
- New classes to ease running Rexx code on GUI threads
  - *AbstractGUIThread*, *AwtGUIThread*, *FXGUIThread*
  - e.g. method *runLater(target, msg, ...)*

BSF4ooRexx 6.41 Going GA

Some Important Changes, 3

- New class *Slot.Argument* (a subclass of *Directory*)
  - The BSF4ooRexx *slot* argument is now an instance of *Slot.Argument* to ease identifying the *slot* argument
  - Added (always last argument) by BSF4ooRexx if ooRexx method got invoked by Java
- New routine *bsf.compile(className,javaCode)*
- New entry *.bsf4rexx~**display.version***
  - Includes versions of ooRexx, BSF4ooRexx and Java

BSF4ooRexx 6.41 Going GA

Some Important Changes, 4

- New routine *ppJavaExceptionChain(co,bTrace)*
  - If a Java exception occurs, BSF4ooRexx raises a Rexx condition containing the Java exception object
    - Java exceptions may be part of a Java *exception chain* such that it may not be clear what the original cause of a Java exception was
      - This routine will display the *Java exception chain* to help identify the original Java exception
    - Arguments
      - *co*: the ooRexx condition object
      - *bTrace*: if *.true* will show stack trace of original Java exception

BSF4ooRexx 6.41 Going GA

Some Important Changes, 5

- Many more changes and additions have taken place
  - E.g. IBM mainframes as a new supported platform
- Consult the text file *changesBSF4ooRexx.txt*
  - Brief list of changes
  - Started 2005, currently 5700 lines
  - More comments may be found in the source code

BSF4ooRexx 6.41 Going GA

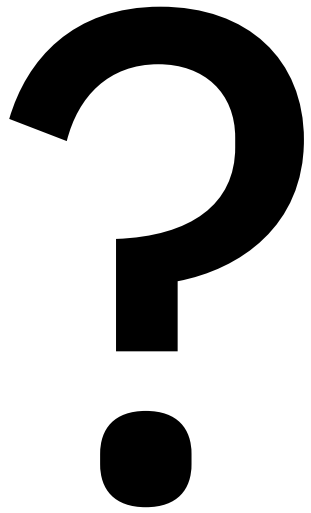
Planned Release Date

- Original plan was to go GA with ooRexx 5.0
- Now: BSF4ooRexx 6.41
  - Currently in gamma
    - No more new features added
    - Just bug fixes, if any bugs reported
  - Planned release: mid of *January 2022*

Roundup and Outlook

- BSF4ooRexx 6.41
  - Since 2014 in the works, many enhancements
  - ooRexx 5.0 beta already in release quality, hence developing on that platform
    - Still attempting to be compatible with ooRexx 4.1
- Next version of BSF4ooRexx
  - Planned version number: 8.50
    - Minimum Java: 8 LTS (2014, supported up to 2030)
    - Minimum ooRexx: 5.0
      - Adding ooRexx 5.0 specific features
      - Other fixes, features may be backported, if needed

Questions ?



Links

- 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/>
  - Introduction to ooRexx on Windows, Slides ("Business Programming 1")
    - <http://wi.wu.ac.at/rgf/wu/lehre/autowin/material/foils/>
- BSF4ooRexx on Sourceforge (ooRexx-Java bridge)
  <https://sourceforge.net/projects/bsf4oorexx/>
  - Introduction to BSF4ooRexx (Windows, Mac, Unix), Slides ("Business Programming 2")
    - <http://wi.wu.ac.at/rgf/wu/lehre/autojava/material/foils/>
- Student's work, including ooRexx, BSF4ooRexx
  <http://wi.wu.ac.at/rgf/diplomarbeiten/>
- JetBrains "IntelliJ IDEA", powerful IDE for all operating systems
  - <https://www.jetbrains.com/idea/download>, free "Community-Edition"
    - Students and lecturers can use the professional edition for free
  - Alexander Seik's ooRexx-Plugin with readme (as of: 2021-11-07)
    - <https://sourceforge.net/projects/bsf4oorexx/files/Sandbox/aseik/ooRexxIDEA/GA/2.0.4/>
- Introduction to ooRexx (254 pages, covers ooRexx 4.2)
  <https://www.facultas.at/Flatscher>