"The Augsburg Version of BSF4Rexx"

2003 International Rexx Symposium

Raleigh, NC (May 2003)

Rony G. Flatscher (Rony.Flatscher@wu-wien.ac.at)
University of Augsburg, Germany (http://www.wiwi.uni-augsburg.de/wi3)
[On leave from the Wirtschaftsuniversität ("vey-uh") Wien (Vienna),
Austria (http://www.wu-wien.ac.at)]

Agenda

- Revealing the *real* Title
- Brief Architecture
  - The "Essener" Version of BSF4Rexx (2001)
  - The "Augsburg" Version of BSF4Rexx (2003)
- An example
  - A Java program
  - A Rexx program
- Additional new features
- Roundup

The Largest External Function Package for Rexx on Earth !

[And already ported to
all important
operating systems and hardware platforms!]

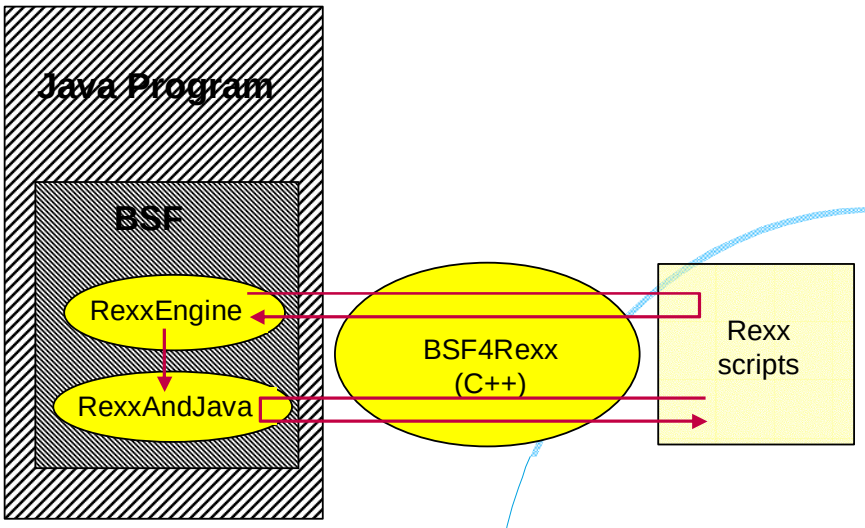
Brief Architecture BSF

- Bean Scripting Framework
  - Open source project of IBM
    - E.g. used in IBM products like WebSphere (JSP)
    - Handed over to the Apache organization in 2002
  - Framework for allowing Java programs to call (non-Java) scripting languages
    - Defines Interfaces for calling back into Java
    - Java objects are stored in a repository on the Java side

Brief Architecture
The Essener Version of BSF4Rexx, 1

- Adds Rexx support
  - Adds an Object Rexx wrapper class
    - "**`BSF.cls`**"
      - Almost makes Java look like Object Rexx
      - Allows sending Object Rexx messages to Java objects
      - Allows using Java arrays as if they were Object Rexx arrays
  - Allows Java programs to invoke Rexx programs
    - Invoked Rexx programs are able to interact with Java objects

Brief Architecture
The Essener Version of BSF4Rexx, 2



Brief Architecture
The Essener Version of BSF4Rexx, 3

- One needs to program Java
  - Rexx programs must be invoked via Java
  - Stub to invoke Rexx programs via Java
    - "**`com.ibm.bsf.Main`**"
      - A crook
    - "True" Rexx programmers are shied away, because they need to know too much about Java
  - **Nevertheless**: allows a true, easy to use and powerful scripting language to script (automate, remote-control) all Java applications!

Brief Architecture
An Example of Java Invoking Rexx

```
import com.ibm.bsf.*;   // BSF support
import java.io.*;       // exception handling

public class TestSimpleExec {

  public static void main (String[] args) throws IOException
  {
    try {
      BSFManager mgr  = new BSFManager ();
      BSFEngine  rexx = mgr.loadScriptingEngine("rexx");
      String     rexxCode = "SAY 'Rexx was here!'";

      rexx.exec ("rexx", 0, 0, rexxCode);

    } catch (BSFException e) { e.printStackTrace(); }
  }
}
```

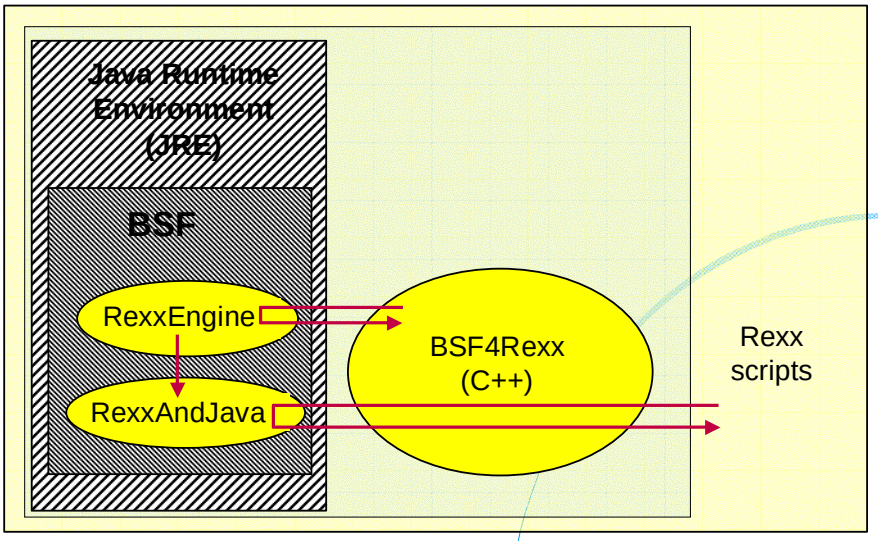
Brief Architecture
The Augsburg Version of BSF4Rexx, 1

- 100 % Compatible to the Essener version
- Adds ability to invoke Java ***from Rexx*** !
  - – No need to be a Java expert
    - Be able to study the HTML documentation of the Java classes
      - – *Everyone* can do that!

      `http://java.sun.com/api`

      - – All Java documentation online via the web!
  - – Able to use *any Java class* and *any Java object* !

Brief Architecture
The Augsburg Version of BSF4Rexx, 2



Brief Architecture
The Augsburg Version of BSF4Rexx, 3

- Removes memory handling bugs in "**BSF4Rexx.cc**"
- Using Mark Hessling's "RexxTrans"
  - One binary of "**BSF4Rexx.cc**" for all supported Rexx interpreters
- Adds external BSF functions
  - Starting and stopping the JVM
    - "**BSFLoadJava**", "**BSFUnloadJava**"
  - Querying all and the registered BSF-functions
    - "**BSFQueryAllFunctions**", "**BSFQueryRegisteredFunctions**"
  - Querying how BSF4Rexx got invoked, via Java or via Rexx
    - "**BSFInvokedBy**"
  - Querying the version of "**BSF4Rexx.cc**"
    - "**BSFVersion**"

Brief Architecture
The Augsburg Version of BSF4Rexx, 4

- Removes entirely the restriction of the maximum of five dimensions on Java arrays
- Pre-registers the most important Java classes in the BSF-registry
  - `'Class.class', 'Object.class', 'Method.class', 'Array.class', 'String.class', 'System.class'`
  - `'boolean.class', 'Boolean.class', 'byte.class', 'Byte.class', 'char.class', 'Character.class', 'double.class', 'Double.class', 'float.class', 'Float.class', 'int.class', 'Integer.class', 'long.class', 'Long.class', 'short.class', 'Short.class', 'void.class', 'Void.class'`

Brief Architecture
The Augsburg Version of BSF4Rexx, 5

- The BSF-registry
  - Maintained by Java
    - Instance of class "**BSFManager**"
    - A directory of those Java objects which can be referred to by Rexx (or any other program in the system) via a unique String value (a "**proxy**")
  - Can be used to interchange data with
    - Java programs,
    - Other Rexx programs (invoked maybe later!) and/or
    - Other scripting languages

An Example
A Java Class "XyzType", 1

```
public class XyzType    // example class for demonstrating BSF4Rexx
{
        // constructors of this class (same name as class!)
    public XyzType () {          // constructor without arguments
        counter=counter+1;      // increase counter
    }

    public XyzType (String initialValue) {   // constructor with argument
        this();                 // invoke (call) constructor withou argument
        info=initialValue;      // save initial value
    }
        // keyword "static": class fields (attributes) and class methods
    static public int counter=0;    // field: will count # of instances


        // instance fields (attributes) and instance methods
    private String info = null;     // field: no value per default

    public String getInfo () {  // accessor (getter) method (function)
        return info;            // return whatever "info" points to
    }

    public void setInfo (String aValue) {   // setter method (function)
        info=aValue;            // save received value with "info"
    }
}
```

An Example
A Java Class "XyzType", 2

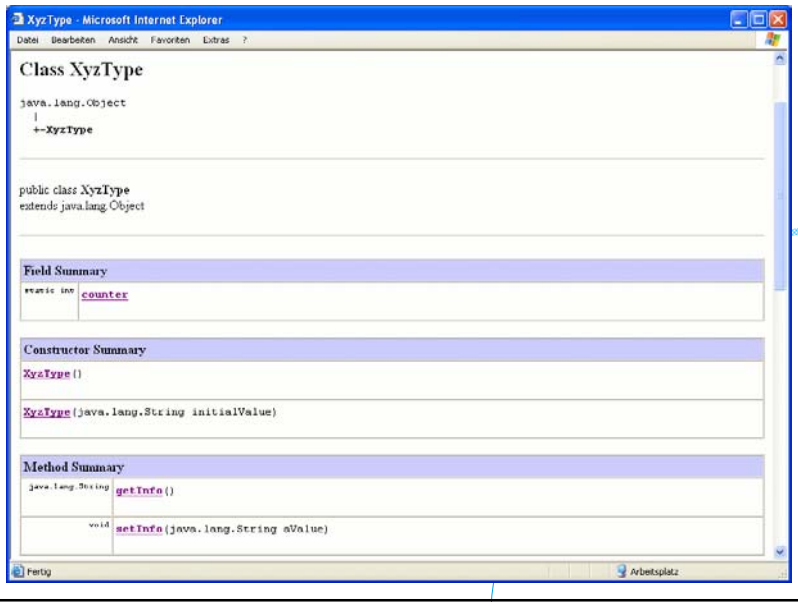
- Save to file "**`XyzType.java`**"
  - Care for case of filename !
    - Must be exactly typed the same as the name of the Java class
- Compile using the Java compiler

```
javac XyzType.java
```

- Create HTML documentation

```
javadoc XyzType
```

An Example
A Java Class "XyzType", 3



An Example
A Rexx Program, 1

```
if rxFuncQuery("BSF") = 1 then   /* BSF() support not loaded yet ? */
do
   call rxFuncAdd "BsfLoadFuncs", "BSF4Rexx", "BsfLoadFuncs"
   call BsfLoadFuncs     /* register the BSF* external functions   */
   call BsfLoadJava      /* load Java, we need it!                 */
end

null=".NIL"              /* representation for Java's "null"       */
javaClass = "XyzType"    /* determine Java class to use            */
   /* query value of static field (attribute) "counter" via class itself */
say "value of static field 'counter'=" || bsf("getStaticValue", javaClass, "counter")
say

   /* creating an instance of the Java class "XyzType"   */
o=BSF("registerBean", null, javaClass)    /* create an instance of "XyzType"     */
say "o:" o
say "# 1:" bsf("invoke", o, "getInfo")    /* get the value via the getter method */
   /* use the object's setter method to define a string value      */
call bsf "invoke", o, "setInfo", "string", "Hello, from Rexx..."
say "# 2:" bsf("invoke", o, "getInfo")    /* get the value via the getter method */
   /* query value of static field (attribute) "counter"  */
say "value of static field 'counter'=" || bsf("getFieldValue", o, "counter")
say

   /* release (unregister) reference to the Java object  */
call BSF "unregisterBean", o              /* remove register entry from Java     */
```

An Example
A Rexx Program, 2

```
... Continued ...

/* create a second Java object      */
say "creating another instance of XyzType, this time with an initial value..."

        /* create an instance of "XyzType" and supply a string value   */
o=BSF("registerBean", "otl", javaClass, "String", "Hi, RexxLA!")
say "o:" o
say "# 3:" bsf("invoke", o, "getInfo")    /* get the value via the getter method */

   /* query value of static field (attribute) "counter" via object  */
say "value of static field 'counter'=" || bsf("getFieldValue", o, "counter")
```

An Example
A Rexx Program (Output), 3

```
value of static field 'counter'=0

o: XyzType@15f5897
# 1: .NIL
# 2: Hello, from Rexx...
value of static field 'counter'=1

creating another instance of XyzType, this time with an initial value...
o: otl
# 3: Hi, RexxLA!
value of static field 'counter'=2
```

An Example
An Object Rexx Program, 1

```
javaClass = "XyzType"   /* determine Java class to use            */
say "value of static field 'counter'=" || .bsf~getStaticValue(javaClass, "counter")
say

o=.BSF~new(javaClass)   /* create an instance of "XyzType"     */
say "o:" o
say "# 1:" o~getInfo    /* get the value via the getter method */
o~setInfo("string", "Hello, from Rexx...")
say "# 2:" o~getInfo    /* get the value via the getter method */
say "value of static field 'counter'=" || o~bsf.getFieldValue("counter")
say
   /* release (unregister) reference to the Java object  */
-- not necessary for Object Rexx: garbage collection will take care of this !!!

   /* create a second Java object      */
say "creating another instance of XyzType, this time with an initial value..."
        /* create an instance of "XyzType" and supply a string value   */
o=.BSF~new(javaClass, "String", "Hi, RexxLA!")
say "o:" o
say "# 3:" o~getInfo    /* get the value via the getter method */
say "value of static field 'counter'=" || o~bsf.getFieldValue("counter")
::requires "BSF.cls"    -- get Object Rexx support
```

An Example
An Object Rexx Program (Output), 2

```
value of static field 'counter'=0

o: XyzType@15f5897
# 1: The NIL object
# 2: Hello, from Rexx...
value of static field 'counter'=1

creating another instance of XyzType, this time with an initial value...
o: XyzType@f9f9d8
# 3: Hi, RexxLA!
value of static field 'counter'=2
```

Roundup

- The Augsburg version of "BSF4Rexx"
  - Full Rexx and Object Rexx language support
  - Open source, free (LGPL license)
  - Allows using
    - All Java classes and objects from Rexx, hence
    - Java runtime environment (JRE) on all platforms
  - Fully portable Rexx and Object Rexx programs *using Java as a single huge "function package"*
  - Discussion, support: **`news:comp.lang.rexx`**
  - URL: **`http://sourceforge.net/projects/bsf4rexx`**
    - [Older URL: **`http://wi.wu-wien.ac.at/rgf/rexx/bsf4rexx]`**
  - Outlook: Apache Jakarta BSF