

2010 REXxLA International Rexx Language Symposium Proceedings

René Vincent Jansen (ed.)

THE REXX LANGUAGE ASSOCIATION
REXXLA Symposium Proceedings Series
ISSN 1534-8954

Publication Data

©Copyright The Rexx Language Association, 2024

All original material in this publication is published under the Creative Commons - Share Alike 3.0 License as stated at <https://creativecommons.org/licenses/by-nc-sa/3.0/us/legalcode>.

A publication of **RexxLA Press**

The responsible publisher of this edition is identified as *IBizz IT Services and Consultancy*, Amsteldijk 14, 1074 HR Amsterdam, a registered company governed by the laws of the Kingdom of The Netherlands.

The RexxLA Symposium Series is registered under ISSN 1534-8954

The 2010 edition is registered under ISBN 978-94-037-2967-1



2024-01-07 First printing

Introduction

History of the International REXX Language Symposium

In 1990, Cathie Dager of SLAC¹ convened the organizing committee for the first independent REXX² Symposium for Developers and Users. SLAC continued to organize this annual event until the middle of the 1990's when the REXXLA took over that responsibility. Symposia have been held annually since 1990.

About REXXLA

During the 1993 Symposium in La Jolla, California, plans for a REXX User Group materialized. The REXX Language Association (REXXLA), as it was called, is an independent, non-profit organization dedicated to promoting the use and understanding of the REXX programming language. REXXLA manages several open source implementations of REXX.

The selection procedure

Presentation proposals are solicited yearly using a CFP³ procedure, after which the REXXLA symposium committee reviews them and votes which presentations are selected for the symposium. The presentations are peer reviewed before being presented. Presenters are not compensated for their presentations.

Location

The 2010 symposium was held in Amsterdam and Almere, The Netherlands from 11 Dec 2010 to 14 Dec 2010.

¹Stanford Linear Accelerator Center, since 2008 SLAC National Accelerator Laboratory

²Cowlshaw, M. F., **The REXX Language** (second edition), ISBN 0-13-780651-5, Prentice-Hall, 1990.

³Call For Papers.

Contents

1	Regina New Functionality / GUI Version of THE – Mark Hessling	1
2	The 2010 Edition of BSF4Rexx – Rony G. Flatscher	5
3	The CSVStream Class – Jon Wolfers	24
4	Building NetRexx Systems – René Vincent Jansen	37
5	Use Rexx and ooRexx from its Java-based Sibling NetRexx – Rony G. Flatscher	53
6	Debugging JDBC Connections using NetRexx – Robert J. Wilson	66
7	Rexx/CSV and Rexx/PDF – Mark Hessling	83
8	PP - the Program Porting Machine – Thomas Schneider	89
9	Orde! with Rexx – Michiel van Hoorn	100
10	System Rexx – Harris M. Morgenstern	116

Regina New Functionality / GUI Version of THE – Mark Hessling

Date and Time

13 Dec 2010, 07:30:00 CET

Presenter

Mark Hessling

Presenter Details

Mark has significant involvement in several REXX-related Open Source projects including the Regina REXX interpreter and THE (The Hessling Editor). Mark is the author of several REXX external function packages, such as REXX/SQL, REXX/DW, REXX/gd, REXX/CURL, REXX/Curses, REXX/PDF, REXX/Wrapper, REXX/WS and REXX/JSON. Mark is also the current principal maintainer of The Regina REXX Interpreter.

Session Abstract

Mark discusses recent developments in the latest versions of his toolset for REXX.

News on Existing Products

2010 International Rexx Symposium
Mark Hessling

13 December 2010
Amsterdam and Almere
The Netherlands



Updates to Old Products

- Regina News
 - V 3.6
- THE News
 - A "real" GUI Version
- Other packages
 - Support for ooRexx 4.x



Regina News

- V3.5 now packaged as .deb
 - Available for Debian and Ubuntu 10.10+
- New platforms
 - Haiku, Minix, Android (unsighted)
- Large file support on 32bit platforms
 - STREAM and SYSFILETREE



THE News

- Windows GUI version
 - New PDCurses port; Win32a – Bill Gray
 - Native GUI Window
 - No Drag 'n' Drop
- Started Unicode Support
 - Time consuming



Other Products News

- ooRexx 4.x support
 - "New" interpreter even though API looks similar
 - 10 packages require updating
- Packaging problems for 64bit platforms
 - Rexx/Trans no longer an option
- New packages
 - Rexx/CSV and Rexx/PDF
 - Other presentation



The 2010 Edition of BSF4Rexx – Rony G. Flatscher

Date and Time

13 Dec 2010, 09:00:00 CET

Presenter

Rony G. Flatscher

Presenter Details

Rony works as a professor for Business informatics (“Wirtschaftsinformatik”) at the Vienna University of Economics and Business Administration (Wirtschaftsuniversität Wien) and uses Open Object REXX for teaching Business Administration and MIS students the object-oriented paradigm, as well as remote-controlling (automating) Windows and Windows end-user applications (e.g. MS Office, Open Office) as well as Java and Java applications (he is the author of BSF4ooREXX, the ooREXX-Java bridge, which uses Apache BSF and had Rony invited to become an ASF member). He consults and trains in all of his research fields.




"The 2010 Edition of BSF4ooRexx"

2010 International Rexx Symposium
Amsterdam/Almere, Netherlands
(December 2010)

© 2010 Rony G. Flatscher (Rony.Flatscher@wu.ac.at)
Wirtschaftsuniversität Wien, Austria (<http://www.wu.ac.at>)



BSF4ooRexx

- 
- External Rexx function package
 - Allows to interact with the Java runtime environment (JRE)
 - Exploit functionality of Java classes
 - Exploit functionality of Java objects
 - ooRexx 4.0.1 and later
 - Package "BSF.CLS"
 - Camouflages Java as ooRexx
 - Supplies class **BSF** and public routines
 - "Everything that is available in Java becomes directly available to ooRexx !"

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]
```

Things to Know About Java, 1

- Strictly typed language
 - Primitive types
 - boolean, byte, char, short, int, long, float, double
 - Object-oriented types
 - Any Java class, e.g.
 - java.awt.Dimension, java.lang.String, java.lang.System, ...
 - Wrapper classes for primitive types
 - java.lang.Boolean, java.lang.Byte, java.lang.Character, java.lang.Short, java.lang.Integer, java.lang.Long, java.lang.Float, java.lang.Double
 - "Boxing": wraps up a primitive value into a wrapper object
 - "Unboxing": retrieves a primitive value from its wrapper object

▼ Things to Know About Java, 2

- Case sensitive
 - Upper- and lowercase significant!
- Classes organized in packages
 - Package names may be compound
 - E.g. "java.lang"
 - Fully "qualified class name" includes package name
 - e.g. "java.lang.String"
 - "Unqualified class name"
 - e.g. "String"

▼ Things to Know About Java, 3

- A Java class may consist of
 - Fields (comparable to ooRexx attributes) and
 - Methods (comparable to ooRexx methods)
- Fields and methods
 - Static fields and static methods
 - Sometimes dubbed "class fields" and "class methods"
 - Available to the class object and its instances
 - Otherwise "instance methods"
 - Only available to instances of a Java class

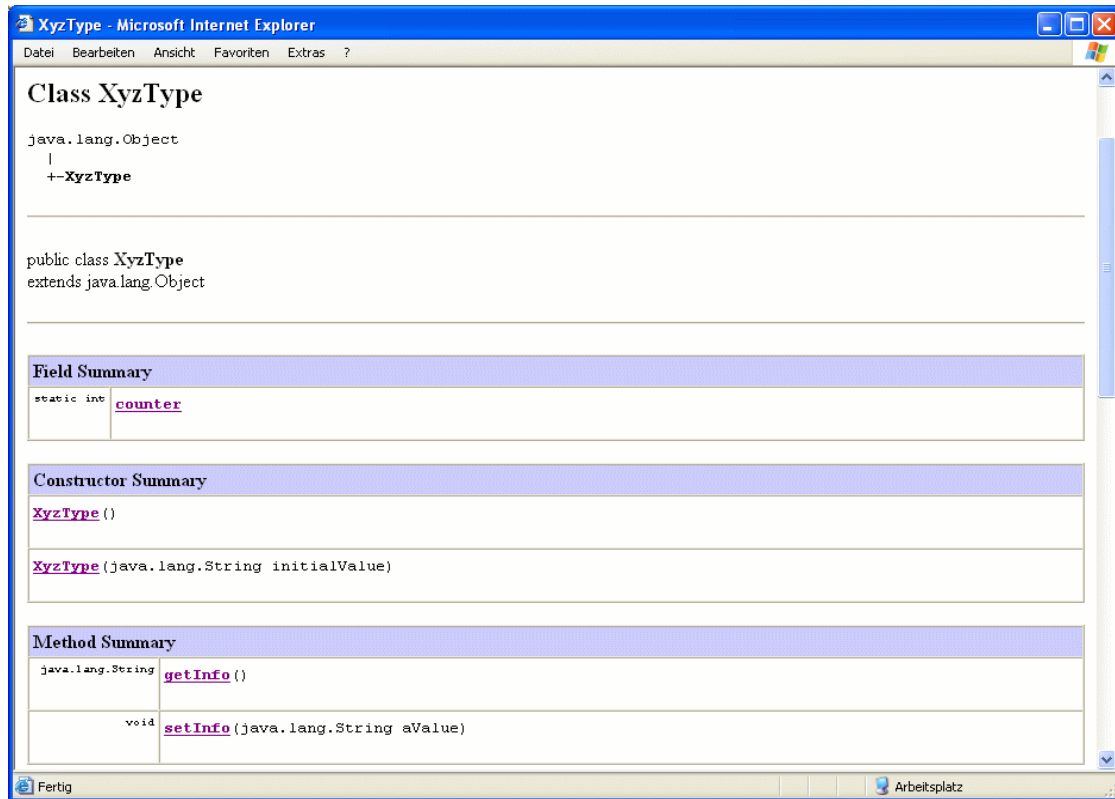
▼ Things to Know About Java, 4

- A Java class, its fields and methods may be
 - "public"
 - These can be accessed by the "world" (everyone)
 - "private"
 - Only accessible within the Java class
 - "protected"
 - Only accessible within Java classes of the same package and subclasses
 - None of the above modifier given
 - Only accessible within Java classes of the same package, but to no one else

▼ Things to Know About Java, 5

- Excellent documentation ("Javadoc")
 - Extensive set of interlinked HTML documents
 - Created right from the comments in Java sources
 - Can be studied on the Internet
 - **First** start out with the older version Java 1.4
 - Overview:
<http://download.oracle.com/javase/1.4.2/docs/index.html>
 - All Java classes:
<http://download.oracle.com/javase/1.4.2/docs/api/>
 - **Then** study the latest version, e.g. Java 1.6 ("Java 6")
 - <http://download.oracle.com/javase/6/docs/index.html>
 - <http://download.oracle.com/javase/6/docs/api/>

▼ A Javadoc Example



▼ BSF.CLS

Camouflages Java as ooRexx

- ooRexx class "**BSF**"
 - Allows to create Java objects
 - Needs at least fully qualified Java class name
- Invoking Java methods
 - Just send the name of the method to the Java object
 - Supply the arguments as documented, if any
 - Type conversions between ooRexx and Java are done automatically by BSF4ooRexx, if necessary
 - Return values are automatically converted by BSF4ooRexx, if necessary



BSF4ooRexx

Example Using Java Class "XYZType", 1

```
o=.BSF~new("XYZType")  
say "o~getInfo:" o~getInfo  
o~setInfo("Hello, from ooRexx...")  
say "o~getInfo:" o~getInfo  
::requires BSF.CLS -- get Java support
```

Output:

```
o~getInfo: The NIL object  
o~getInfo: Hello, from ooRexx...
```



BSF.CLS

Camouflages Java as ooRexx

- ooRexx class "**BSF**" (continued)
 - Allows to create Java objects
 - Needs at least fully qualified Java class name
- Possible arguments for *creating* Java objects
 - Can be found by studying the "Constructor" section in the Javadocs
 - Supply the arguments as documented after the fully qualified Java class name argument
 - Type conversions between ooRexx and Java are done automatically by BSF4ooRexx, if necessary



BSF4ooRexx

Example Using Java Class "XYZType", 2

```
o=.BSF~new("XYZType", "This value was supplied at Java object creation.")  
say "o~getInfo:" o~getInfo  
o~setInfo("Hello, from ooRexx...")  
say "o~getInfo:" o~getInfo  
::requires BSF.CLS -- get Java support
```

Output:

```
o~getInfo: This value was supplied at Java object creation.  
o~getInfo: Hello, from ooRexx...
```



BSF.CLS

Camouflages Java as ooRexx

- Allows to import any Java class
 - `bsf.import(JavaClassName)`
 - Java class name
 - Use of the exact case is mandatory !
 - Java class name must be fully qualified !
 - Imported class can be treated as if it was an ooRexx class
 - Allows to use the ooRexx "new"-method to create instances of the imported Java class
 - Possible arguments for creating Java objects can be found by studying the "Constructor" section in the Javadocs