

# 2009 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 2009 edition is registered under ISBN 978-94-037-2976-3



2024-01-07 First printing

---

## Introduction

### History of the International REXX Language Symposium

In 1990, Cathie Dager of SLAC<sup>1</sup> convened the organizing committee for the first independent REXX<sup>2</sup> 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 CFP<sup>3</sup> 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 2009 symposium was held in Chilworth Manor, Winchester, UK from 17 May 2009 to 21 May 2009.

### Organizing Committee

- David Ashley
- Gil Barmwater
- Jon Wolfers
- Mark Hessling
- René Jansen
- Rony G. Flatscher

---

<sup>1</sup>Stanford Linear Accelerator Center, since 2008 SLAC National Accelerator Laboratory

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

<sup>3</sup>Call For Papers.

---

# Contents

|    |  |     |
|----|--|-----|
| 1  | ooRexx 4.0 New Function – Gil Barmwater                                      | 1   |
| 2  | Building ooRexx 4.00 on MacOSX – René Vincent Jansen                         | 14  |
| 3  | The 2009 Edition of BSF4Rexx Part I – Rony G. Flatscher                      | 26  |
| 4  | The 2009 Edition of BSF4Rexx Part II – Rony G. Flatscher                     | 44  |
| 5  | An Object Rexx Retrospective – Simon Nash                                    | 53  |
| 6  | Rexx Open Source Business Update – Virgil Hein                               | 62  |
| 7  | Rexx the Data Converter – Mike Cowlshaw                                      | 68  |
| 8  | CGI Programming Your Website – Steve Swift                                   | 82  |
| 9  | History of the Rexx ANSI standardization effort – Brian Marks                | 91  |
| 10 | Tool Updates – Mark Hessling   | 95  |
| 11 | Future Netrex – René Vincent Jansen  | 99  |
| 12 | Using Rexx to teach secure server programming by example – Mike Protts       | 120 |
| 13 | Totalising Tables and Streaming Databases - Subclassing ooRexx – Jon Wolfers | 123 |
| 14 | Building a Cross-Platform GUI App using Rexx/DW – Mark Hessling              | 146 |
| 15 | rgf_util2.rex – Rony G. Flatscher  | 150 |

# ooRexx 4.0 New Function – Gil Barmwater

## Date and Time

18 May 2009, 09:30:00 CET

## Presenter

Gil Barmwater

## Presenter Details

Gil Barmwater is a retired independent computer consultant who has used REXX since it first became available within IBM. In addition to programming REXX in the mainframe environment, he has developed REXX solutions on OS/2 and multiple versions of Windows. Learning about the features of ooREXX and making contributions to its ongoing development is his current interest. He previously served as the Secretary/Treasurer and Vice President of the REXX Language Association.

## Session Abstract

A foray into the new function offered by the ooRexx 4.0 version

---

---

# Open Object Rexx 4.0.0

## What's New?



Gil Barmwater  
(gil\_b@bellsouth.net)  
Retired Consultant

---

---

## Overview

- New Platforms
- New BIFs
- New Directives
- New Classes
- New Methods
- Other New Items
- Current Status

---

---

# New Platforms

- Support for both types of OS
  - 32-bit
  - 64-bit
- Built and tested on
  - Windows
  - Linux
  - AIX
  - Mac (soon)

---

---

## New Platforms (2)

- Extensive code rewrite required
  - needed to permit 64-bit compilation
  - allowed some internal redesign
  - enhances maintainability and extensibility
- Additional external APIs
  - object oriented vs. string only
  - all ooRexx supplied external function libraries upgraded
  - original APIs still available

---

---

## New Platforms (3)

- Documentation restructured
  - Rexx Reference
    - Windows-specific classes removed
  - New Windows Extensions manual
  - Windows OODialog Reference
    - be sure to read sect. 1.2 & 2.4
    - new and enhanced classes and objects
    - some items deprecated

---

---

## New Platforms (4)

- Rxapi
  - Now runs as a system wide daemon on \*NIX (service on Windows)
  - Uses sockets vs shared memory for communication
- Windows Installation
  - uninstall only removes files installed by the ooRexx package
  - WSH support temporarily disabled
  - .REX files are now "drop handlers"

---

---

## New Platforms (5)

- REXX\_PATH
  - added to Path
  - used for program searches
  - set by user

---

---

## New BIFs

- Qualify()
  - ANSI standard function
  - Returns the fully qualified file name for *name*
    - e.g. `qualify(name)`
    - merely adds directory info.

## New BIFs (enhancements)

- Filespec()
  - New options
    - Location - Drive + Path
    - Extension
- Time()
  - New option
    - Offset - returns the offset of the local time from UTC in microseconds
    - e.g. TIME("O")

## New BIFs (more enhancements)

- Lastpos() & Pos()
  - New parameter
    - length
    - e.g. LASTPOS(" ", "abc def ghi", 7, 3) -> 0
- Translate()
  - New parameters
    - pos, length
    - e.g. TRANSLATE("abcdef", , , , 2, 3) -> "aBCDef"

---

---

# New Directives

- IBM Object Rexx had 4
  - CLASS, METHOD, REQUIRES, ROUTINE
- ooRexx 3.2 added 1
  - ATTRIBUTE
- ooRexx 4.0.0 adds 2 more
  - CONSTANT
    - e.g. ::constant e 2.718
  - OPTIONS
    - e.g. :options digits 20 trace O

---

---

# New Classes

- Fundamental Classes
  - Package
    - an encapsulation for an ooRexx source file
    - new mechanism for external functions i.e. RxFuncADD
    - dynamic ::REQUIRES
  - Routine
    - call and callWith methods to run
    - dynamic ::ROUTINE
    - build executable code w/o INTERPRET

---

---

## New Classes (2)

- Collection Classes

- IdentityTable

- like Table but index object must be the same instance on searches
    - used mostly for situations where you need to keep track of actual object instances rather than relying on "equality", which may not be the same
    - interpreter uses this internally

---

---

## New Classes (3)

- Utility Classes

- Orderable

- MIXIN class that provides all of the comparison methods like  $<$ ,  $=$ , etc.

- Weak Reference

- Use to maintain caches of objects without preventing them from being reclaimed by the garbage collector.

---

---

## New Classes (4)

- Utility Classes Related to New APIs
  - Buffer, Pointer, RexxContext
    - cannot be created from Rexx code
  - Buffer
    - a Rexx interpreter managed block of storage
  - Pointer
    - a wrapper around a native pointer value

---

---

## New Classes (5)

- Utility Classes Related to New APIs (2)
  - RexxContext
    - gives access to context information about the currently executing Rexx code
    - use `.CONTEXT` to access
      - ▶ `.context~digits` -> same as `digits()`
      - ▶ `.context~variables` -> a directory of the variables in the current execution context
      - ▶ `.context~package`

---

---

# New Methods

- Object Class

- send
- sendWith
- startWith

---

---

# New Methods (2)

- DateTime Class

- offset
- toTimeZone
- utcDate
- utcIsoDate
- Class methods

- support timezone offset when creating new instances

- Method Class

- package

---

---

## New Methods (3)

- MutableBuffer Class
  - many more of the methods of the String class
- OLEObject Class
  - addEventMethod, removeEventMethod
  - connectEvents, disconnectEvents
  - isConnectable
  - isConnected
  - removeEventHandler

---

---

## New Methods (4)

- RexxQueue Class
  - empty
  - makeArray
  - exists
  - open
- Stem Class
  - toDirectory
- String Class
  - replaceAt
    - "abcdef"~replaceAt(" ",3, 3) -> "ab f"

---

---

## Other New Items

- Utility Class
  - sockets.cls
    - needs `::REQUIRES` to use
- Samples
  - sockets
  - ooDialog and Windows system
- ooRexxTry
  - GUI (ooDialog) version of RexxTry

---

---

## Other New Items (2)

- Miscellaneous Enhancements
  - EXTERNAL keyword on `::ATTRIBUTE`, `::METHOD`, `::ROUTINE`
  - LIBRARY keyword on `::REQUIRES`
  - String length > 999,999,999 on any 64-bit OS
  - Stream Class supports files > 4GB

---

---

## Other New Items (3)

- Program Behavior Changes
  - `::REQUIRES` only loads once
  - Negative RCs returned correctly
  - `::ATTRIBUTE` now correctly GUARDED
  - `RxFuncAdd` now verifies resolution
  - Returns from `RxMath` functions now consistent
  - Private Methods now also accessible from other instances of the same class and the class itself

---

---

## Current Status

- In Beta test
  - latest refresh built yesterday
- Release Candidate
  - 22 May 2009
- Target Release
  - 8 June 2009

# Building ooRexx 4.00 on MacOSX – René Vincent Jansen

## Date and Time

18 May 2009, 11:00:00 CET

## Presenter

René Vincent Jansen

## Presenter Details

René has used REXX since it appeared in TSO Extensions in the second half of the eighties when he was a systems programmer at the Central Bank of The Netherlands. He is an independent consultant since the turn of the century, specializing in models and meta models in order to rationalize data governance and model driven development. He likes to program in any language as long as it's REXX.