2022 RexxLA International Rexx Language Symposium Proceedings

René Vincent Jansen (ed.)

10 May 2023

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

Publication Data

©Copyright The Rexx Language Association, 2023

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 2022 edition is registered under ISBN 978-94-036-4146-1

ISSN 1534-8954

9 771534 895004 >

ISBN 978-94-036-4146-1

9 789403 641461 >

2023-03-14 First printing

Introduction

History of the International Rexx Language Symposium

In 1990, Cathie Dager of SLAC[1] convened the organizing committee for the first independent Rexx[2] 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[3] procedure, after which the RexxLA symposium comittee 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 2022 symposium was held Online from 12 Sep 2022 to 14 Sep 2022.

Organizing Committee

- Chip Davis
- Jon Wolfers
- Mark Hessling
- René Jansen
- Terry Fuller

[1] Stanford Linear Accelerator Center, since 2008 SLAC National Accelerator Laboratory
[2] Cowlishaw, M. F., **The REXX Language** (second edition), ISBN 0-13-780651-5, Prentice-Hall, 1990.
[3] Call For Papers.

Contents

1 NetRexx 4.04 GA update – René Vincent Jansen **1**

2 Rexx, Python and Flutter experiences from an EU project – Michael Beer **15**

3 The New Interactive Trace in NetRexx – Marc Remes **33**

4 CREXX Level B MVP Release – Adrian Sutherland **41**

5 Hidden Corners of Regina – Mark Hessling **53**

6 BSF4ooRexx: From 641 GA Update to 850 Beta – Rony G. Flatscher **64**

7 BSF4ooRexx 850 Beta: Exploiting ooRexx 5.0 Direct and Redirecting Command Environment Handlers Implemented in Java/NetRexx – Rony G. Flatscher **76**

8 Linux Tiling Window Managers with ooRexx – Till Winkler **92**

9 Jenkins - Current state of affairs – P.O. Jonsson **106**

10 Introducing the JDOR Rexx Command Handler for Easy Creation of Bitmaps and Bitmap Manipulations on Windows, Mac and Linux – Rony G. Flatscher **113**

11 30 Years of THE – Mark Hessling **130**

12 Setting Up Endeavour OS Linux for ARM on the Raspberry Pi 4 – Tony Dycks **143**

13 Hacks, Tips and Tricks to Install NetRexx 4 and BSF4ooRexx on the RPi 4 Using Oracle Linux 8 – Tony Dycks **162**

14 Building multi-platform Docker container images for Rexx implementations – René Vincent Jansen **179**

15 Extracting large amounts of data efficiently from excel using ooRexx – Jon Wolfers **191**

1

NetRexx 4.04 GA update – René Vincent Jansen

Date and Time

12 Sep 2022, 18:00:00 CET

Presenter

René Vincent Jansen

Presenter Details

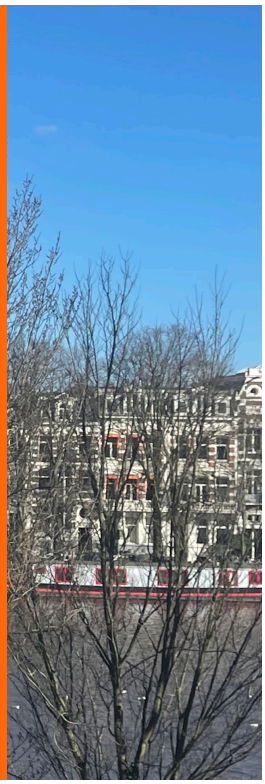
René has used Rexx since it appeared in TSO Extensions in the second half of the eighties when he was a systems programmer at the Central Bank of The Netherlands. He is an independent consultant since the turn of the century, specializing in models and meta models in order to rationalize data governance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

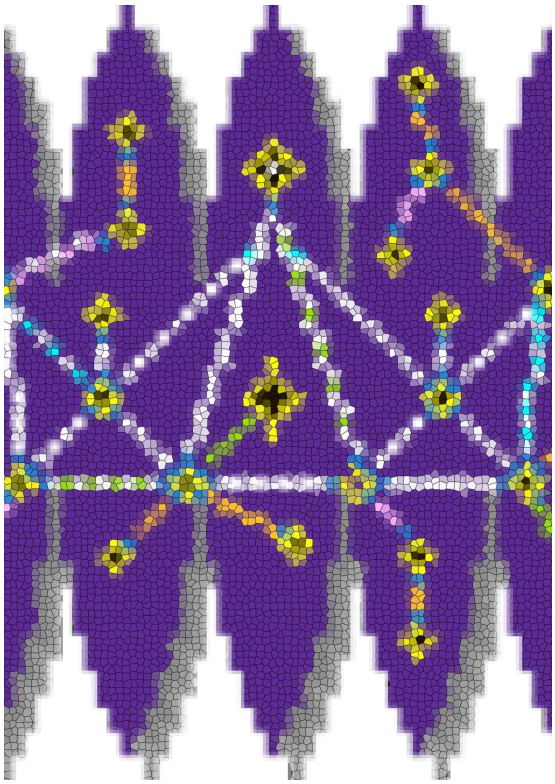
Session Abstract

NetRexx 4.0.4 should be GA by the first day of the symposium. It is future proof and more compatible with the past than ever. Some of the new functionality (and some added in 4.0.3) will be presented.

NetRexx 4.04 GA

René Vincent Jansen
33rd Rexx Language Symposium
12 September 2022





NetRexx 4.04

Our 4.04 release is more **future proof**, and at the same time more **compatible with the past** than it ever was

Agenda

1 Project Developments

2 Translator Developments

3 Workspace Developments

4 Pipelines Developments

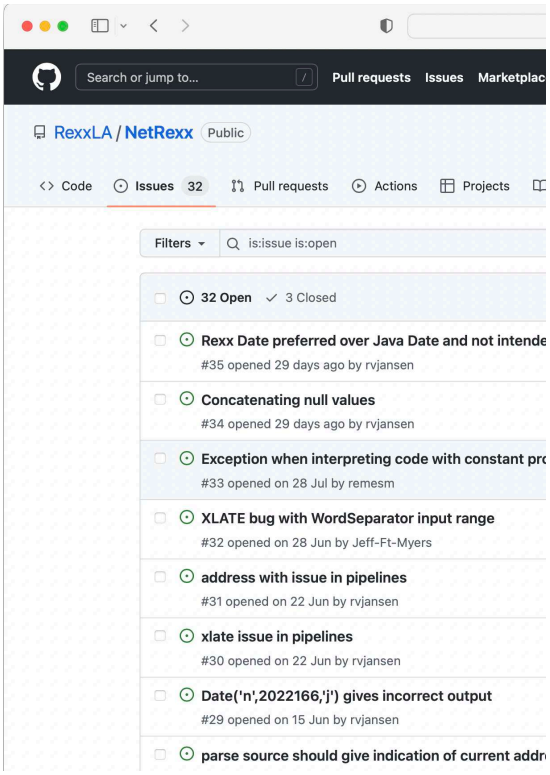
5 Documentation and Examples

№ 1 Project Developments

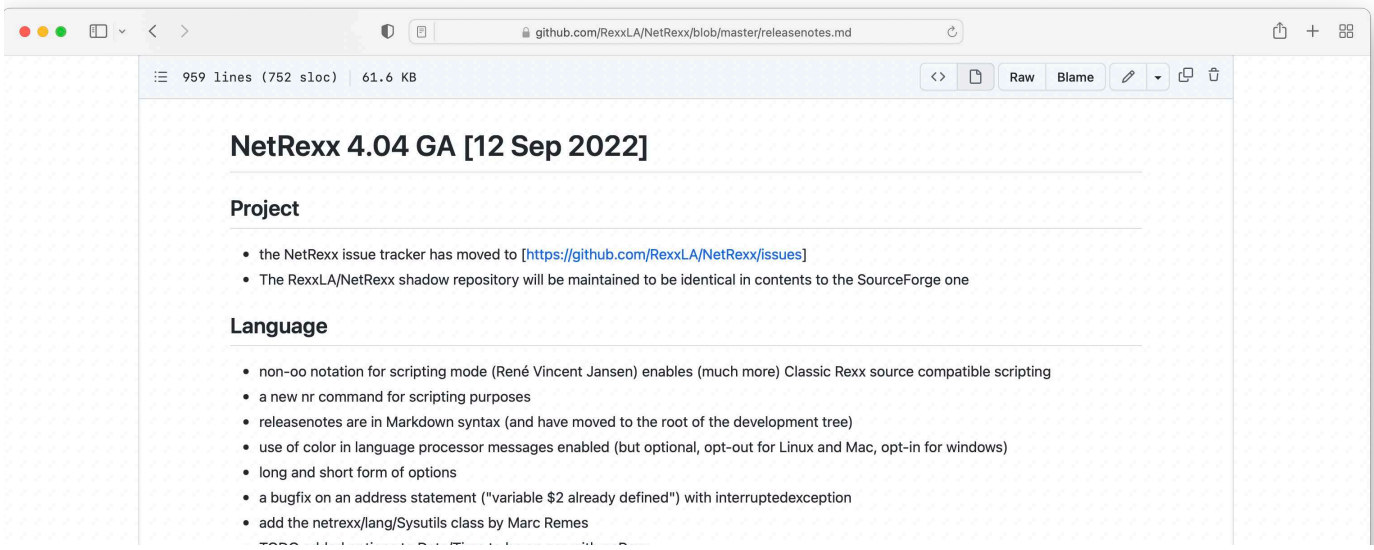
5

Project Location

- Issue List has moved to Github
  - After being in disuse since Oracle shut down our previous forge
  - SourceForge has a finite future for the RexxLA NetRexx project
    - Among other reasons, instability and taking our list addresses hostage
  - https://github.com/RexxLA/NetRexx/issues
- This repository is also a shadow location for the source code
  - https://github.com/RexxLA/NetRexx



6



The release notes are in Markdown now

№ 2

Translator Developments

Classic Rexx compatible BIF

- BIF: Built in Functions (substr(), length(), left(), right() *etcetera*.
- Implemented in the RexxRexx class (to be analogous with RexxDate, RexxStream and other compatibility efforts)
  - Automatically included in ‘Scripting Mode’
  - In ‘class mode’: use ‘uses’

```
File Edit Options Buffers Tools Index NetRexx Help
package netrexx.lang
options binary decimal strictargs strictcase noformat
/**
 * With this one weird trick ... Rexx is back in 1984
 * Class RexxRexx provides static versions of the Rexx
 * to be called without oo notation.
 * Created on: Fri, 01, Apr 2022 13:04:19 +0200
 */
class RexxRexx

  method RexxRexx()

  method abbrev(a=Rexx, b=Rexx, length='') static
    if length='' then len=b.length()
    else len=length
    return a.abbrev(b,len)

  method abs(a) static returns Rexx signals NumberForm
    return a.abs()

  method b2d(a=Rexx,dil=Rexx -1) static returns Rexx
    return b2d(a,dil)

  method b2x(a=Rexx) static returns Rexx
    return a.b2x()

  method centre(a=Rexx, wid=Rexx, pad=Rexx ' ') static
    return a.centre(wid,pad)

  method center(a=Rexx,wid=Rexx, pad=Rexx ' ') static
    return a.center(wid,pad)

  method changestr(a=Rexx,old=Rexx,new=Rexx) static re
    return old.changestr(a,new)
```

Why? “It really was fine as it was …”

- NetRexx was conceived as a fully object-oriented language, and method calls have the form ***[instance].[method](parameters)***
  - The instance points to where the non-oo (let’s call it procedural) version has the object that needs to be processed by the function
- Over the years, users indicated that they would like to use older scripts that they had written over the years
  - ‘*Scripting mode*’ (available from the early beginnings) is a nod to that requirement
- Users of simple pipelines especially balked at having to switch around the objects and arguments in their own pipeline stages - these users are only in it for the (mostly compatible) pipelines, not specifically the other NetRexx benefits
- This increases your chance of running an old (but really simple) script unchanged,
  - And greatly reduces the work to get a one-off running quickly
- (And the ultimate retort - you don’t have to use it if you don’t want to)

```
file: half.rexx
/* rexx */
parse arg number
if datatype(number,'whole')
then say number%2 + number //2
else say ' argument is not a whole number'
```


Static functions

- These implementations of the BIF’s are **static** calls
  - Ironically, the greatest benefit is for those that are a bit unsure about the difference between instance methods and static methods
  - You cannot do the usual oo-magic here, but that is the whole point

```
File Edit Options Buffers Tools Index NetRexx Help
package netrexx.lang
options binary decimal strictargs strictcase noformat
/**
 * With this one weird trick ... Rexx is back in 1984
 * Class RexxRexx provides static versions of the Rexx
 * to be called without oo notation.
 * Created on: Fri, 01, Apr 2022 13:04:19 +0200
 */
class RexxRexx

  method RexxRexx()

  method abbrev(a=Rexx, b=Rexx, length='') static
    if length='' then len=b.length()
    else len=length
    return a.abbrev(b,len)

  method abs(a) static returns Rexx signals NumberForm
    return a.abs()

  method b2d(a=Rexx,dil=Rexx -1) static returns Rexx
    return b2d(a,dil)

  method b2x(a=Rexx) static returns Rexx
    return a.b2x()

  method centre(a=Rexx, wid=Rexx, pad=Rexx ' ') static
    return a.centre(wid,pad)

  method center(a=Rexx,wid=Rexx, pad=Rexx ' ') static
    return a.center(wid,pad)

  method changestr(a=Rexx,old=Rexx,new=Rexx) static re
    return old.changestr(a,new)
```

Use of Colour/Color

- The translator now uses ANSI colour sequences to indicate the difference between warnings and errors
- This is switched on as a default, except for Windows, where we need to wait until more people use a sensible shell
- There are options **-color and -nocolor**
  - When -colour and -nocolour are really needed, we'll see the votes on the issue and will do our homework
-

Some bug fixes

- a bugfix on an address statement ("variable $2 already defined") with interruptedexception
- date('j') input format gave incorrect output and is fixed. This version 'century windows' around 75 for compatibility with z/VM CMS Rexx; in a coming release 2-digit years will be deprecated on date'j' input format.

Sysutils

- The SysUtils class by Marc Remes is added to the distribution
  - SysUtils is a traditional package that delivers
    - sleep()
    - I/O Functions
      - File search - search a path
      - File properties (datetime)
      - Create/remove directories
      - SysFileTree
    - Mutex Semaphores
    - Event Semaphores

NetRexx Trace Interactive

This deserves its own presentation, which Marc Remes will deliver

TRACE INT

Marc Remes

remesm@gmail.com

№ 3

NetRexx Workspace Developments

What is new and/or fixed in the Workspace

- You can now exit it with the **exit** command
  - I know this isn't a great feat but you don't have to remember …. what was that again …?
- The **timer** on the sessions has been fixed
- It now starts up with the timer option (right side of screen) as a default
  - This was the intention earlier but that first starting time was way off - so instead you were shown the session number (which was correct)
- the default prompt changed to 'Ready;' (but both options can still be changed in an $HOME/nrws.properties file)
- the **pipelines processor** is now loaded by default

№ 4 Pipelines Developments

Pipelines

- a pipelines example that uses the ADDRESS WITH functionality in a NetRexx script
- Stage **locate** can be abbreviate to locat, loc, lo and l

```
OUT =''
ADDRESS PIPE WITH OUTPUT STEM OUT
"LITERAL AAP NOOT MIES | SPLIT | REVERSE | CONS"
DO I=1 TO OUT[0]
  SAY OUT[I]
END
```

№ 5

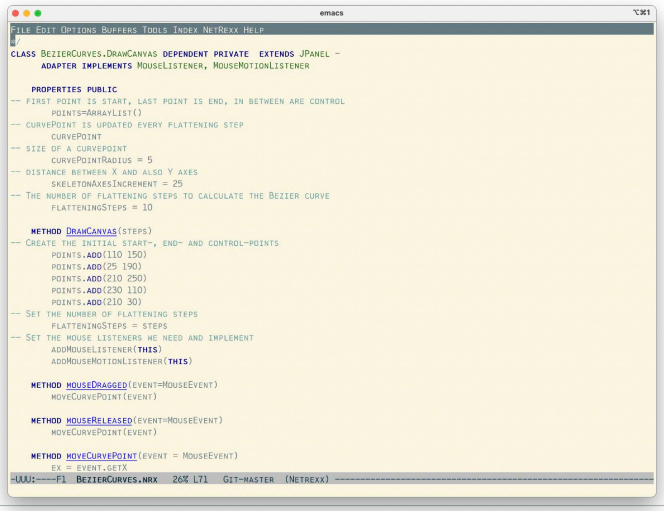
Documentation and Examples

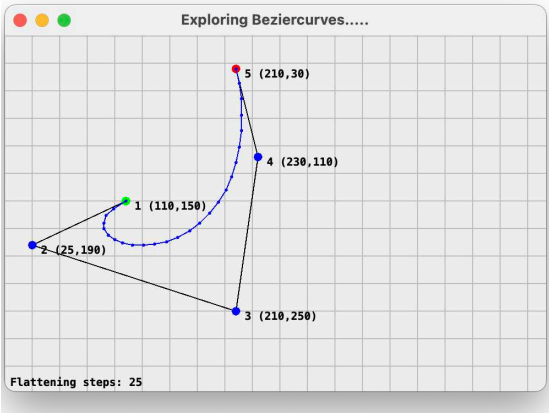
Documentation

- Programmer's Guide made up to date with currently included scripting environments
- New Date(), Time(), and Stream() (linein, line out, charin, charout) functions documented
- ADDRESS WITH documentation added
-

A wonderful Swing GUI Sample

- Thanks to Ruurd Idenburg





№ 6 Recap of the 4.03 changes

(4.03 was released after the 2021 Symposium)

NetRexx 4.03 changes

- - ADDRESS WITH implemented (Marc Remes)
- - a fix for NETREXX-24 (was 148): ITERATE without label is not inside LOOP construct message (Marc Remes)
- - a small fix for the MOD option (Jason Martin)
- - correct slightly erroneous DataIO2.nrx example from NrxRedBk (René Vincent Jansen) (Thank you Terry Fuller)
- - a fix for JavaFX usage with the JPMS (and any other imported module) (Marc Remes) (Thank you Terry Fuller)
- - a new sample how to address https by Ruurd J. Idenburg
- - A Classic Rexx inspired Stream I/O implementation (René V. Jansen)
- (Thank you Leslie Turiff for testing and issue reports)
- - A new example of how to use a Swing Gui by Ruurd J. Idenburg
- - Start of a new organisation of the examples directory by function
- - a 'scripting mode' application adds 'uses RexxDate, RexxStream' to the generated class definition
- - RxJrt now compiles under Java 8 and RxJrtApi.nrx under Java >=9 (Marc Remes)
- - The java2nrx tool has been upgraded to v1.05
- - The distribution package now also contains java2nrx (previously only in source package)
- - in scripting mode, 'parse pull' can be used as an equivalent of 'parse ask'

№ 7

Next: 4.05

NetRexx 4.05

- **Your suggestions, please**
- Documentation check and update

- Might contain
  - Closer integration of Rexx and String types
  - Calling natively implemented Rexx libraries

  - Thank you for your attention!

2

Rexx, Python and Flutter experiences from an EU project – Michael Beer

Date and Time

12 Sep 2022, 18:15:00 CET

Presenter

Michael Beer

Presenter Details

CEO and Founder of mindcoa.ch. Focus on providing psychological online help with AI and validated scientific procedures. Experience in IT, TelCo, Pharma/Health, Media, Tourism, Automotive, Finance. Previous roles include global sales and marketing functions (e.g. IBM). Decades of application development projects with several programming languages including Rexx and APL. Large parts of the mindcoa.ch framework are written in Regina, using services of IBM Watson, Microsoft Azure and other providers.

Session Abstract

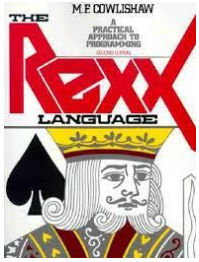
Within the EU AAL framework a stress management solution solution for employees 50+ was developed. The user works with an android-based app, VR-glasses and a device measuring physical stress parameters (HRV). Several programming languages/environments and different backend systems were involved. The presentation covers the system design and special challenges for REXX.

REXX, Python and Flutter

Experiences from an EU Project

M. Beer, 2022

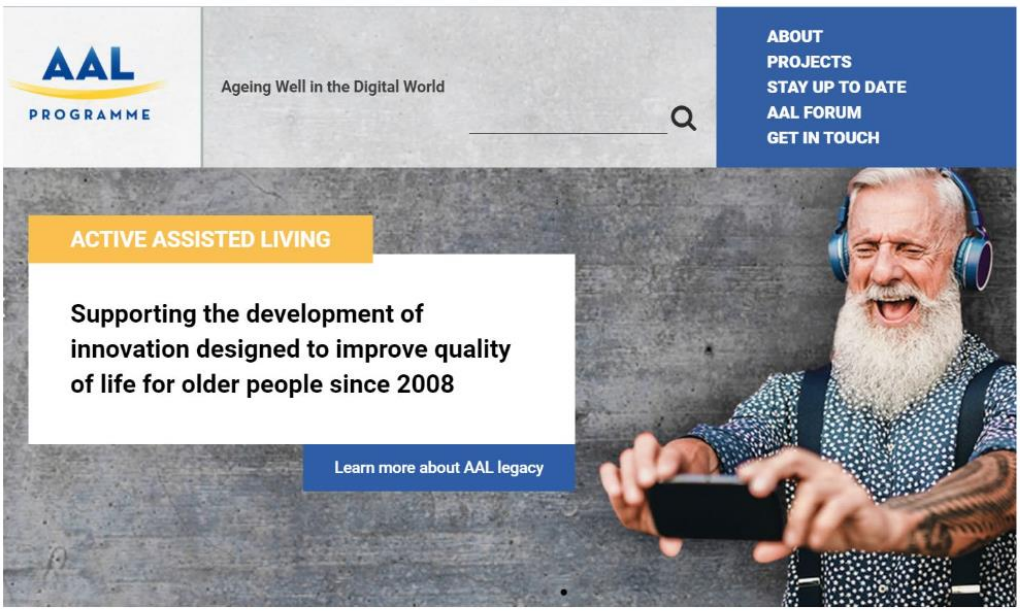
mike@mindcoa.ch



Agenda

- Project Context
- Components
- Functional Overview
- Development Process & Tools
- Challenges

Project Context



AAL is a European programme funding innovation that keeps us connected, healthy, active and happy into our old age.

AAL Programme supports the development of products and services that make a real difference to people's lives – for those facing some of the challenges of ageing and for those who care for older people if they need help.

http://www.aal-europe.eu