

ooRexx 5.0 New Features

Amsterdam, 2017-04-12



Rony G. Flatscher

Agenda

- Download URLs
- Brief overview
 - Compared to ooRexx 4.2
 - Concepts
- Examples for some of the new features
 - ooRexx nutshell examples
- Roundup

Download URLs

- State of ooRexx 5.0 as of 2017-04-12
 - Beta
 - Stable
 - Available for Windows, Linux and MacOSX (via [BSF4ooRexx](#))
- Download URLs
 - <https://sourceforge.net/projects/oorex/5.0.0beta/>
 - <https://sourceforge.net/projects/bsf4oorex/files/beta/20161026/>
 - Lookout for the latest beta versions (current April 2017)

Brief Overview, 1

- New raildiagrams in documentation
- Added namespace ability

namespace:publicRoutine

namespace:publicClass

- New directives

::ANNOTATE

- Add (meta-)information to Rexx packages, routines, classes, methods, methods, constants

::RESOURCE

- Add multiline text blocks

Brief Overview, 2

- New Rexx classes
 - AlarmNotification
 - MessageNotification
 - RexxInfo
 - StringTable
 - Ticker
- New environment symbols
 - .resources
 - .RexxInfo

Brief Overview, 3

- Added syntax sugar for arrays (new notation)
 - Any comma delimited values in an expression yield an array
(element1, element2, ... elementN)
- Keyword instructions with new subkeywords
 - DO WITH INDEX** idx **ITEM** value **OVER** collection **FOR** times
 - LOOP WITH INDEX** idx **ITEM** value **OVER** collection **FOR** times
 - SELECT CASE** expression
 - USE LOCAL**
 - New invocation indicator **REQUIRES** in **PARSE SOURCE**

Nutshell Example, New Array Notation, 1

```
loop val over "one", "deux", 3, "quattro"  
  say val  
end  
say "----"
```

```
value="one", "deux", 3, "quattro"  
loop val over value  
  say val  
end  
say "----"
```

```
say "nr. of items:" ("one", "deux", 3, "quattro")~items
```

Output:

```
one  
deux  
3  
quattro  
---  
one  
deux  
3  
quattro  
---  
nr. of items: 4
```

Nutshell Example, New Array Notation, 2

```
loop with index idx item val over "one", "deux", 3, "quattro"  
    say "idx="idx "val="val  
end  
say "----"
```

```
values="one", "deux", 3, "quattro"  
loop with index idx item val over values for 2  
    say "idx="idx "val="val  
end
```

Output:

```
idx=1 val=one  
idx=2 val=deux  
idx=3 val=3  
idx=4 val=quattro  
---  
idx=1 val=one  
idx=2 val=deux
```


Nutshell Example, **SELECT CASE**

```
select case .dateTime~new~weekday
  when 1 then say "Monday"
  when 2 then say "Tuesday"
  when 3 then say "Wednesday"
  when 4 then say "Thursday"
  when 5 then say "Friday"
  otherwise say "weekend"
end
```

Output (run on 2017-04-11):

Tuesday

Nutshell Example, **::ANNOTATE** Directive, 1

```
call prettyPrint "package:",      .context~package~annotations
call prettyPrint "prettyPrint:",  .routines~prettyPrint~annotations
call prettyPrint ".test:",        .test~annotations
```

```
::annotate package occasion "Rexx Symposium 2017 Amsterdam" author "Rony G. Flatscher"
::annotate package test "demonstrates there are three annotations for package"
```

```
::routine prettyPrint
  use arg title, dir
  say title
  do idx over dir
    say "      ["idx"]:" "["dir~at(idx)"]"
  end
  return "["arg1"]"
```

```
::annotate routine prettyPrint purpose "enclose square brackets around argument"
```

```
::class test
```

```
::annotate class test author "RGF"
```

```
::method hi
```

```
  say "this is a demo method to allow to be annotated"
```

```
::annotate method hi purpose "demo annotations"
```

Nutshell Example, **::ANNOTATE** Directive, 2

Output:

package:

```
[TEST]: [demonstrates there are three annotations for package]
```

```
[OCCASION]: [Rexx Symposium 2017 Amsterdam]
```

```
[AUTHOR]: [Rony G. Flatscher]
```

prettyPrint:

```
[PURPOSE]: [enclose square brackets around argument]
```

.test:

```
[AUTHOR]: [RGF]
```

Nutshell Example, **::RESOURCE** Directive, 1

```
say .resources~copyright
say "----"
say .resources~htmlCopyright
say "----"
```

```
::resource copyright
    This is my personal copyright that I came up with
    while creating the slides to fill-in the talk about
    new features in the upcoming ooRexx 5.0...
::END
```

```
::resource htmlCopyright end myend
    <h1>My Copyright</h1>

    <p> This is my personal copyright that I came up with
    while creating the slides to fill-in the talk about
    new features in the upcoming ooRexx 5.0...
    (can even use
::END
    in here)
MYEND
```

Nutshell Example, **::RESOURCE** Directive, 2

Output:

```
This is my personal copyright that I came up with  
while creating the slides to fill-in the talk about  
new features in the upcoming ooRexx 5.0...
```

```
---
```

```
<h1>My Copyright</h1>
```

```
<p> This is my personal copyright that I came up with  
while creating the slides to fill-in the talk about  
new features in the upcoming ooRexx 5.0...
```

```
(can even use
```

```
::END
```

```
in here)
```

```
---
```

Nutshell Example, **USE LOCAL**

```
o=.test~new("Mike F.", "Cowlshaw", "1234-456")
say o

::class test          -- a test class
::attribute firstname
::attribute lastname
::attribute phoneNumber
::attribute greetings

::method init        -- constructor: use attributes by default
  use local tmpStr   -- variable "tmpStr" is NOT an attribute!
  use arg firstname, lastname, phoneNumber
  tmpStr=firstname lastname
  greetings="Dear" tmpStr

::method string      -- use attributes by default
  use local
  return firstName";"lastname"." phoneNumber"." greetings
```

Output:

```
Mike F.;Cowlshaw. 1234-456. Dear Mike F. Cowlshaw
```

Nutshell Example, **.RexxInfo** Class, 1

```
do info over "architecture", "caseSensitiveFiles", "digits", "internalDigits"  
  say info:" .rexxinfo~send(info)  
end
```

```
do info over "internalMinNumber", "internalMaxNumber", "package", "platform"  
  say info:" .rexxinfo~send(info)  
end
```

```
do info over "release", "version", "revision", -  
  "languageLevel", "maxExponent", "maxPathLength"  
  say info:" .rexxinfo~send(info)  
end
```

Nutshell Example, **.RexxInfo** Class, 2

Output:

```
architecture: 64
caseSensitiveFiles: 1
digits: 9
internalDigits: 18
internalMinNumber: -9999999999999999999
internalMaxNumber: 9999999999999999999
package: The REXX Package
platform: DARWIN
release: 0
version: 5.0.0
revision: 0
languageLevel: 6.05
maxExponent: 999999999
maxPathLength: 1024
```


Roundup and Outlook

- ooRexx 5.0
 - Fully backwardly compatible
 - Rexx
 - ooRexx
 - Many useful additions to the programming language
 - Currently in beta
 - Stable beta
 - Please test thoroughly and report any problems you encounter
 - Enter report in the ooRexx Sourceforge "Bug Tracker"!
 - No date for general availability ("GA") announced yet