The Release of ooRexx 5.2.0

Changes and New Features Since 5.1.0

37th International Rexx Symposium
May 3rd – May 6th 2026, Barcelona

Prof. Rony G. Flatscher

Overview

- Changes and additions in the context of ::OPTIONS
- Updates of .File, .Json, .TraceObject classes
- New .Yaml class
- New built-in function gc()
- Incorporating "utf8proc" library
- Fixes and updates in the context of multithreading

Prof. Rony G. Flatscher

::OPTIONS, 1

- Classic Rexx
  - Changing trace and numeric settings take effect for the entire program
- ooRexx
  - Rexx code defined for ::routine and ::method directives define their own scope
    - E.g., labels are only visible within the scope
    - Trace and numeric settings are confined to the scope
  - Note: in ooRexx the code before the first directive is called *prolog*
    - Defines its own scope
  - The static ::OPTIONS directive allows to define global settings
    - See Rexx Reference (rexxref.pdf), "3.6. ::OPTIONS"

::OPTIONS, 2

Default Values

- ::OPTIONS

  DIGITS 9
  FORM SCIENTIFIC
  FUZZ 0
  NUMERIC NOINHERIT (new in 5.2.0)
  ERROR CONDITION
  FAILURE CONDITION
  LOSTDIGITS CONDITION
  NOSTRING CONDITION
  NOTREADY CONDITION
  NOVALUE CONDITION
  PROLOG
  TRACE NORMAL

Prof. Rony G. Flatscher

::OPTIONS
NUMERIC {NOINHERIT | INHERIT}, 1

- New suboption
  - **NUMERIC NOINHERIT** (default)
  - **NUMERIC INHERIT**
    - Called routines will inherit the numeric settings of the caller
      - NUMERIC DIGITS
      - NUMERIC FORM
      - NUMERIC FUZZ

::OPTIONS

NUMERIC {NOINHERIT | INHERIT}, 2

```
-- test_classic_01.rex
say "digits: " digits() "1/17:" 1/17
numeric digits 15
say "digits: " digits() "1/17:" 1/17
call work1
call work3  -- external program "work3.rex"
call work2
exit

work1:
   say "work1: " "digits:" digits() "1/17:" 1/17
   return

work2: procedure
   say "work2: " "digits:" digits() "1/17:" 1/17
   return
```

```
-- work3.rex
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

Output:

```
digits:  9 1/17: 0.0588235294
digits:  15 1/17: 0.0588235294117647
work1:  digits: 15 1/17: 0.0588235294117647
work3*: digits: 9 1/17: 0.0588235294
work2:  digits: 15 1/17: 0.0588235294117647
```


Prof. Rony G. Flatscher

::OPTIONS
NUMERIC {NOINHERIT | INHERIT}, 3

```
-- test_orx_01.rex
say "digits: " digits() "1/17:" 1/17
numeric digits 15
say "digits: " digits() "1/17:" 1/17
call work1
call work3  -- external program "work3.rex"
call work2
exit

::routine work1
   say "work1: " "digits:" digits() "1/17:" 1/17
   return

::routine work2
   say "work2: " "digits:" digits() "1/17:" 1/17
   return
```

```
-- work3.rex
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

Output:

```
digits:  9 1/17: 0.0588235294
digits:  15 1/17: 0.058823529411764 7
work1:  digits: 9 1/17: 0.0588235294
work3*: digits: 9 1/17: 0.0588235294
work2:  digits: 9 1/17: 0.0588235294
```


Prof. Rony G. Flatscher

::OPTIONS
NUMERIC {NOINHERIT | INHERIT}, 4

```
-- test_classic_02.rex
say "digits: " digits() "1/17:" 1/17
numeric digits 15
say "digits: " digits() "1/17:" 1/17
call work1
call work3  -- external program "work3.rex"
call work4  -- external program "work4.rex"
call work2
exit

work1:
   say "work1: " "digits:" digits() "1/17:" 1/17
   return

work2: procedure
   say "work2: " "digits:" digits() "1/17:" 1/17
   return
```

```
-- work3.rex
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

```
-- work4.rex
say "work4*:" "digits:" digits() "1/17:" 1/17
return

::options numeric inherit
```

Output:

```
digits:  9 1/17: 0.0588235294
digits:  15 1/17: 0.0588235294117647
work1:  digits: 15 1/17: 0.0588235294117647
work3*: digits: 9 1/17: 0.0588235294
work4*: digits: 15 1/17: 0.0588235294117647
work2:  digits: 15 1/17: 0.0588235294117647
```

::OPTIONS
NUMERIC {NOINHERIT | INHERIT}, 5

```
-- test_orx_02.rex
say "digits: " digits() "1/17:" 1/17
numeric digits 15
say "digits: " digits() "1/17:" 1/17
call work1
call work3  -- external program "work3.rex"
call work2
exit

::routine work1
   say "work1: " "digits:" digits() "1/17:" 1/17
   return

::routine work2
   say "work2: " "digits:" digits() "1/17:" 1/17
   return

::options numeric inherit
```

```
-- work3.rex
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

Output:

```
digits:  9 1/17: 0.0588235294
digits:  15 1/17: 0.0588235294117647
work1:  digits: 15 1/17: 0.0588235294117647
work3*: digits: 9 1/17: 0.0588235294
work2:  digits: 15 1/17: 0.0588235294117647
```


Prof. Rony G. Flatscher

::OPTIONS
NUMERIC {NOINHERIT | INHERIT}, 6

```
-- test_orx_02.rex
say "digits: " digits() "1/17:" 1/17
numeric digits 15
say "digits: " digits() "1/17:" 1/17
call work1
call work3  -- external program "work3.rex"
call work2
exit

::routine work1
   say "work1: " "digits:" digits() "1/17:" 1/17
   return

::routine work2
   say "work2: " "digits:" digits() "1/17:" 1/17
   return

::options numeric inherit
```

```
-- work3.rex
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

```
-- work4.rex
say "work4*:" "digits:" digits() "1/17:" 1/17
return

::options numeric inherit
```

Output:

```
digits:  9 1/17: 0.0588235294
digits:  15 1/17: 0.0588235294117647
work1:  digits: 15 1/17: 0.0588235294117647
work3*: digits: 9 1/17: 0.0588235294
work4*: digits: 15 1/17: 0.0588235294117647
work2:  digits: 15 1/17: 0.0588235294117647
```

Package, 1

- An ooRexx program is a *package* that may consist of
  - *Prolog* code ("main program") with its own scope
  - *Directives* which are led in with two consecutive colons (::) *after* the prolog
    - ::ANNOTATE ... allow to annotate the prolog, classes, methods and routines
    - ::ATTRIBUTE ... define an attribute for a class, creates getter and setter methods
    - ::CLASS ... define a class (a.k.a. type, a.k.a. structure)
    - ::CONSTANT ... define a constant value
    - ::METHOD ... define a method and its code for a class, each defines an own scope
    - ::OPTIONS ... define global options for the package
    - ::REQUIRES ... request interpreter to call another Rexx program (package)
    - ::RESOURCE ... define a resource of strings, returned as a string array
    - ::ROUTINE ... define a routine and its code, each defines an own scope

Package, 2

- ooRexx defines a class named **Package** which ooRexx uses at runtime to manage and maintain all relevant information about each program (a "package"), e.g., methods like
  - classes (returns a StringTable of all defined classes), addPublicClass (allows to add a public class at runtime), ...
- A program can get at its package object at runtime using the .context environment symbol

```
thisPkg = .context~package
```

  and analyze or change it dynamically
- Up to ooRexx 5.1.0 it was not possible to find out at runtime which options are in effect for a particular program/package, let alone allowing to change them dynamically, e.g., for debugging
  - ooRexx 5.2.0 added
    - The class method defaultOptions
      - Please note: default options *never* override explicitly set options in a program (package)
    - The instance method options

  to query and change the default options and the options of an individual package (program)

Package Class, 3

- Class method defaultOptions
  - Query and change the default options to use for calling new programs/packages
    - Argument # 1: "**D**[efinedDefaultOptions]"
      - If argument # 2 is omitted, then it returns all current default options formatted as a single "::OPTIONS" string
    - Optional argument # 2: "::OPTIONS" string to define new default options
      - Changes the default options when calling a new program/package later
  - Query and change numeric indicator, which controls whether default options get applied to new called programs/packages
    - Argument # 1: "**C**[ountOverrides]"
      - If argument # 2 is omitted a number gets returned: 0 indicates that default options do not get applied when calling new programs/packages, a positive number n will apply the default options n times, whereas a negative number -n will always apply the default options
    - Optional argument # 2: any whole number

Package Class, 4

- Instance method options
  - Allows to query and change the current package options
    - Argument # 1, option name, if argument # 2 is omitted, returns current value
      - If argument # 1 is "All", then argument # 2 must be supplied with either the value "Condition" or "Syntax" and will set the options named "ERROR", "FAILURE", "LOSTDIGITS", "NOSTRING", "NOTREADY", "NOVALUE"
    - Optional argument # 2, new value for the option named in argument # 1
      - Note: new values only take effect when invoking routines or sending messages the next time

Prof. Rony G. Flatscher

Package Class

Example 1

```
-- test_orx_01.rex
say "digits: " digits() "1/17:" 1/17
numeric digits 15
say "digits: " digits() "1/17:" 1/17
call work1
call work3  -- external program "work3.rex"
call work2
exit

::routine work1
say "options:" .context~package~options
   say "work1: " "digits:" digits() "1/17:" 1/17
   return

::routine work2
   say "work2: " "digits:" digits() "1/17:" 1/17
   return
```

```
-- work3.rex
say "work3*:" .context~package~options
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

Output:

```
digits:  9 1/17: 0.0588235294
digits:  15 1/17: 0.0588235294117647
options: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
work1:  digits: 9 1/17: 0.0588235294
work3*: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
work3*: digits: 9 1/17: 0.0588235294
work2:  digits: 9 1/17: 0.0588235294
```

Package Class

Example 2

```
-- test_orx_02.rex
say "options:" .context~package~options
say "digits: " digits() "1/17:" 1/17
numeric digits 15
say "digits:" digits() "1/17:" 1/17
call work1
call work3  -- external program "work3.rex"
call work2
exit
::routine work1
   say "work1: " "digits:" digits() "1/17:" 1/17
   return
::routine work2
   say "work2: " "digits:" digits() "1/17:" 1/17
   return
::options numeric inherit
```

```
-- work3.rex
say "work3*:" .context~package~options
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

Output:

```
options: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC INHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
digits:  9 1/17: 0.0588235294
digits: 15 1/17: 0.0588235294117647
work1:  digits: 15 1/17: 0.0588235294117647
work3*: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
work3*: digits: 9 1/17: 0.0588235294
work2:  digits: 15 1/17: 0.0588235294117647
```

Package Class

Example 3

```
-- test_orx_03.rex
say "options:" .context~package~options
say "digits: " digits() "1/17:" 1/17
numeric digits 15
say "digits:" digits() "1/17:" 1/17
call work1
call work3  -- external program "work3.rex"
call work4  -- external program "work4.rex"
call work2
exit

::routine work1
   say "work1: " "digits:" digits() "1/17:" 1/17
   return

::routine work2
   say "work2: " "digits:" digits() "1/17:" 1/17
   return
::options numeric inherit
```

```
-- work3.rex
say "work3*:" .context~package~options
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

```
-- work4.rex
say "work4*:" .context~package~options
say "work4*:" "digits:" digits() "1/17:" 1/17
return

::options numeric inherit
```

Output:

```
options: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC INHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
digits:  9 1/17: 0.0588235294
digits: 15 1/17: 0.0588235294117647
work1:  digits: 15 1/17: 0.0588235294117647
work3*: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
work3*: digits: 9 1/17: 0.0588235294
work4*: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC INHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
work4*: digits: 15 1/17: 0.0588235294117647
work2:  digits: 15 1/17: 0.0588235294117647
```

Package Class

Example 4, Programs/Packages

```
-- test_orx_04.rex
pkg = .context~package  -- get this package
object
say "options:" pkg~options
say "digits: " digits() "1/17:" 1/17
pkg~options("digits",15)
pkg~options("trace","result")
say "digits: " digits() "1/17:" 1/17
say "new *internal* invocations (in this package)
will have:"
say pkg~options
say "..."
call work1
call work3  -- external program "work3.rex"
call work4  -- external program "work4.rex"
call work2
exit
::routine work1
   say .context~package~options
   say "work1: " "digits:" digits() "1/17:" 1/17
   return
::routine work2
   say .context~package~options
   say "work2: " "digits:" digits() "1/17:" 1/17
   return
```

```
-- work3.rex
say "work3*:" .context~package~options
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

```
-- work4.rex
say "work4*:" .context~package~options
say "work4*:" "digits:" digits() "1/17:" 1/17
return

::options numeric inherit
```


Prof. Rony G. Flatscher

Package Class

Example 4, Output 1/2

Output:

```
options: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
digits:  9 1/17: 0.0588235294
digits:  9 1/17: 0.0588235294
new *internal* invocations (in this package) will have:
::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION NOSTRING
CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS
...
       >I> Routine "WORK1" in package "C:\rgf\test_orx_04.rex".
    17 *-* say .context~package~options
       >>>   "::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS"
::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION NOSTRING
CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS
    18 *-* say "work1: " "digits:" digits() "1/17:" 1/17
       >>>   "work1:  digits: 15 1/17: 0.0588235294117647"
work1:  digits: 15 1/17: 0.0588235294117647
    19 *-* return
       <I< Routine "WORK1" in package "C:\rgf\test_orx_04.rex".
work3*: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
work3*: digits: 9 1/17: 0.0588235294

 … continued …
```

Package Class

Example 4, Output 2/2

Output:

```
… continued …

work4*: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC INHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
work4*: digits: 9 1/17: 0.0588235294
       >I> Routine "WORK2" in package "C:\rgf\test_orx_04.rex".
    21 *-* say .context~package~options
       >>>   "::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS
CONDITION NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS"
::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION NOSTRING
CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS
    22 *-* say "work2: " "digits:" digits() "1/17:" 1/17
       >>>   "work2:  digits: 15 1/17: 0.0588235294117647"
work2:  digits: 15 1/17: 0.0588235294117647
    23 *-* return
       <I< Routine "WORK2" in package "C:\rgf\test_orx_04.rex".
```

Package Class

Example 5, Programs/Packages

```
-- test_orx_05.rex
say ".package~defaultOptions:" .package~defaultOptions("d")
.package~defaultOptions("D", "::options digits 15 trace r")
.package~defaultOptions("countOverrides", -1)
say "calling programs/packages will have as their default options:"
say ".package~defaultOptions:" .package~defaultOptions("D")
say "digits: " digits() "1/17:" 1/17
say "..."
call work1
call work3  -- external program "work3.rex"
call work4  -- external program "work4.rex"
call work2
exit
::routine work1
   say "work1: " .context~package~options
   say "work1: " "digits:" digits() "1/17:" 1/17
   return
::routine work2
   say "work2: " .context~package~options
   say "work2: " "digits:" digits() "1/17:" 1/17
   return
```

```
-- work3.rex
say "work3*:" .context~package~options
say "work3*:" "digits:" digits() "1/17:" 1/17
return
```

```
-- work4.rex
say "work4*:" .context~package~options
say "work4*:" "digits:" digits() "1/17:" 1/17
return

::options numeric inherit
```


Prof. Rony G. Flatscher

Package Class

Example 5, Output 1/2

Output:

```
.package~defaultOptions: ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS
CONDITION NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
calling programs/packages will have as their default options:
.package~defaultOptions: ::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION
LOSTDIGITS CONDITION NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS
digits:  9 1/17: 0.0588235294
...
work1:  ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
work1:  digits: 9 1/17: 0.0588235294
       >I> Routine "C:\rgf\work3.rex" in package "C:\rgf\work3.rex".
     2 *-* say "work3*:" .context~package~options
       >>>   "work3*: ::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS
CONDITION NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS"
work3*: ::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS
     3 *-* say "work3*:" "digits:" digits() "1/17:" 1/17
       >>>   "work3*: digits: 15 1/17: 0.0588235294117647"
work3*: digits: 15 1/17: 0.0588235294117647
     4 *-* return
       <I< Routine "C:\rgf\work3.rex" in package "C:\rgf\work3.rex".

… continued …
```

Package Class

Example 5, Output 2/2

Output:

```
… continued …

       >I> Routine "C:\rgf\work4.rex" in package "C:\rgf\work4.rex".
     2 *-* say "work4*:" .context~package~options
       >>>   "work4*: ::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC INHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS
CONDITION NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS"
work4*: ::OPTIONS DIGITS 15 FORM SCIENTIFIC FUZZ 0 NUMERIC INHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE RESULTS
     3 *-* say "work4*:" "digits:" digits() "1/17:" 1/17
       >>>   "work4*: digits: 9 1/17: 0.0588235294"
work4*: digits: 9 1/17: 0.0588235294
     4 *-* return
       <I< Routine "C:\rgf\work4.rex" in package "C:\rgf\work4.rex".
work2:  ::OPTIONS DIGITS 9 FORM SCIENTIFIC FUZZ 0 NUMERIC NOINHERIT ERROR CONDITION FAILURE CONDITION LOSTDIGITS CONDITION
NOSTRING CONDITION NOTREADY CONDITION NOVALUE CONDITION PROLOG TRACE NORMAL
work2:  digits: 9 1/17: 0.0588235294
```

New Command Line Option

- New command line option to override program's options

  `rexx[.exe] -o[d] "::OPTIONS …" rexxProgram [arguments]`

  - `-o` … override default options for `rexxProgram` only
  - `-od` … override default opttions for `rexxProgram` and all of its called or required programs
  - `"::OPTIONS …"` any definitions
  - `rexxProgram` … the Rexx program to run

Prof. Rony G. Flatscher

Command Line Options

```
-- test_1.rex
thisFileName=filespec("Name",.context~name)
say "L".line":" thisFileName":" "trace():" trace()"," "digits():" digits()"," "1/17:" 1/17
call "test_2.rex"
```

```
-- test_2.rex
thisFileName=filespec("Name",.context~name)
say "L".line":" thisFileName":" "trace():" trace()"," "digits():" digits()"," "1/17:" 1/17
```

Output:

```
E:\rony>rexx test_1.rex
L3: test_1.rex: trace(): N, digits(): 9, 1/17: 0.0588235294
L3: test_2.rex: trace(): N, digits(): 9, 1/17: 0.0588235294

E:\rony>rexx -o  "::options trace r" test_1.rex
     2 *-* thisFileName=filespec("Name",.context~name)
       >>>   "test_1.rex"
     3 *-* say "L".line":" thisFileName":" "trace():" trace()"," "digits():" digits()"," "1/17:" 1/17
       >>>   "L3: test_1.rex: trace(): R, digits(): 9, 1/17: 0.0588235294"
L3: test_1.rex: trace(): R, digits(): 9, 1/17: 0.0588235294
     4 *-* call "test_2.rex"
L3: test_2.rex: trace(): N, digits(): 9, 1/17: 0.0588235294

E:\rony>rexx -o  "::options digits 24" test_1.rex
L3: test_1.rex: trace(): N, digits(): 24, 1/17: 0.058823529411764705882352
L3: test_2.rex: trace(): N, digits(): 9, 1/17: 0.0588235294

E:\rony>rexx -od "::options digits 24" test_1.rex
L3: test_1.rex: trace(): N, digits(): 24, 1/17: 0.058823529411764705882352
L3: test_2.rex: trace(): N, digits(): 24, 1/17: 0.058823529411764705882352
```

New Class Methods for File Class, 1

- The File class gains new class methods to ease reading and writing entire files
  - Text files
    - `readLines( fileName )` ... reads all text lines into an array and returns it
    - `writeLines( fileName, orderdCollection [, Append | Replace] )` ... writes the content of an ordered collection array into a file, by default in appending mode
  - Binary files
    - `readChars( fileName )` ... reads and returns all bytes as a string
    - `writeChars( fileName, string [, Append | Replace] )` ... writes the bytes of the string into a file, by default in appending mode

New Class Methods for File Class, 2

```
-- demoReadWriteLines.rex
parse arg fname
say "file: '"fname"' SysFileExists:" SysFileExists(fname)
code = .file~readLines(fname)
say "'"fname"' has" code~items "lines"
r = .routine~new("calcCone", code)
say "running '"fname"' via a routine object ..."
r~call("817 9")
newFile = fname"-writeLines.bkp"
say "writing array to '"newFile"' (in replace mode)"
.file~writeLines(newFile, code, "replace")
say
say "--- now using readChars/writeChars"
content = .file~readChars(fname)
say "'"fname"' has a size of" content~length "chars"
newFile = fname"-writeChars.bkp"
say "writing chars to '"newFile"' (in replace mode)"
.file~writeChars(newFile, content, "replace")
```

```
-- calcCones.rex
parse arg startNumber loops
lenLoops = length(loops)
width = 15
numeric digits width
val = startNumber
-- multiplications
op = "*"
do i=2 to loops
   str = val~right(width)
   val = val * i
   say str op i~right(lenLoops)
end
-- divisions
op = "/"
i=2
do i=2 to loops
   str = val~right(width)
   val = val / i
   say str op i~right(lenLoops)
end
say val~right(width)
```


Prof. Rony G. Flatscher

New Class Methods for File Class, 3

```
E:\rony>demoReadWriteLines.rex calcCones.rex
file: 'calcCones.rex' SysFileExists: 1
'calcCones.rex' has 22 lines
running 'calcCones.rex' via a routine object ...
           817 * 2
          1634 * 3
          4902 * 4
         19608 * 5
         98040 * 6
        588240 * 7
       4117680 * 8
      32941440 * 9
     296472960 / 2
     148236480 / 3
      49412160 / 4
      12353040 / 5
       2470608 / 6
        411768 / 7
         58824 / 8
          7353 / 9
           817
writing array to 'calcCones.rex-writeLines.bkp' (in replace mode)

--- now using readChars/writeChars
'calcCones.rex' has a size of 407 chars
writing chars to 'calcCones.rex-writeChars.bkp' (in replace mode)
```

```
E:\rony>dir
… cut …
29.04.2026  21:24               407 calcCones.rex
29.04.2026  21:37               407 calcCones.rex-writeChars.bkp
29.04.2026  21:37               407 calcCones.rex-writeLines.bkp
29.04.2026  21:31               679 demoReadWriteLines.rex
```

Updates to the JSON Class, 1

- Updated to conform to RFC 8259
- Now passes the comprehensive JSON Parsing Test Suite from `https://github.com/nst/JSONTestSuite`
- Cf. "Rexx Extensions Library Reference" (`rexxextensions.pdf`)
- ooRexx MapCollections are encoded as JSON objects
- Decodes JSON objects to ooRexx Directory objects
- Added ability to encode to JSON XML and decode from JSON XML
  - Uses either json.dtd (default) or json.xsd
    - JSON XML encoded files can be transformed to JSON with the supplied xmlToJson.xsl
  - New methods jsonToXml, jsonToXmlFile, parseXml, parseXmlFile

Updates to the JSON Class, 2

```
jsonData = '{"name": "Alice", "age": 30, "active": true}'
say "reading jsonData:" jsonData
obj = .json~fromJSON(jsonData)
say
say "turning Rexx object to JSON-XML:"
xml = .json~jsonToXml(obj)
say xml
say
say "turning JSON-XML into Rexx object:"
obj = .json~new~parseXML(xml)
say 'obj["name"]  :' obj["name"]    -- Alice
say 'obj["age"]   :' obj["age"]     -- 30
say 'obj["active"]:' obj["active"]  -- 1 (a JsonBoolean singleton)
say
say "saving object in a JSON-XML file ..."
fn ="03_json.xml"
.json~jsonToXmlFile(obj, fn)

::requires "json.cls"   -- get JSON support
```

Output:

```
reading jsonData: {"name": "Alice", "age": 30, "active": true}

turning Rexx object to JSON-XML:
<?xml version="1.0" encoding="UTF-8"?>
<json xmlns="urn:json:xml:1.0">
  <object>
    <entry>
      <n>active</n>
      <value>
        <boolean>true</boolean>
      </value>
    </entry>
    <entry>
      <n>age</n>
      <value>
        <number>30</number>
      </value>
    </entry>
    <entry>
      <n>name</n>
      <value>
        <string>Alice</string>
      </value>
    </entry>
  </object>
</json>


turning JSON-XML into Rexx object:
obj["name"]  : Alice
obj["age"]   : 30
obj["active"]: 1

saving object in a JSON-XML file ...
```

Update to the TraceObject Class, 1

- While tracing TraceObject instances get created at the start of each trace at which time an append message gets sent to the collector attribute, if it is not .nil
  - At that point in time not all information of a TraceObject has been filled-in yet
  - If a real-time debugger wishes to inspect the full state of a TraceObject at its creation time, one would need an additional callback when the TraceObject setup got completed
- The new notify attribute will get an append message, if it is not .nil when the TraceObject's setup got completed
  - This allows for full inspection of all TraceObject information by a real-time debugger

Update to the TraceObject Class, 2

```
   -- the collector gets the APPEND message at creation time
.TraceObject~collector=.myCollector~new
   -- the notifier gets the APPEND message upon completion
.TraceObject~notify   =.myNotifier~new

trace results
say "L".line":" "3+4="3+4
say
trace normal

::class  myCollector    -- intended for debugging
::method append         -- invoked at TraceObject creation time
  use arg traceObject
  say "---> myCollector, #" traceObject~number '"'traceObject'"'

::class  myNotifier     -- intended for realtime debugging
::method append
  use arg traceObject   -- invoked upon TraceObject's completion
  say "<--- myNotifier,  #" traceObject~number '"'traceObject'"'
  say "<--->"
```

Output:

```
---> myCollector, # 1 "?"
<--- myNotifier,  # 1 "     7 *-* say "L".line":" "3+4="3+4"
<--->
     7 *-* say "L".line":" "3+4="3+4
---> myCollector, # 2 "?"
<--- myNotifier,  # 2 "       >>>   "L7: 3+4=7""
<--->
       >>>   "L7: 3+4=7"
L7: 3+4=7
---> myCollector, # 3 "?"
<--- myNotifier,  # 3 "     8 *-* say"
<--->
     8 *-* say
---> myCollector, # 4 "?"
<--- myNotifier,  # 4 "       >>>   """
<--->
       >>>   ""

---> myCollector, # 5 "?"
<--- myNotifier,  # 5 "     9 *-* trace normal"
<--->
     9 *-* trace normal
```

New YAML Class, 1

- Comprehensive yaml 1.2 (core) support
  - Cf. `https://en.wikipedia.org/wiki/YAML` and `https://yaml.org/`
- Passes the official comprehensive YAML Test Suite from `https://github.com/yaml/yaml-test-suite`
- Cf. "Rexx Extensions Library Reference" (`rexxextensions.pdf`)
- Supports multi-document files and front-matters
- Decodes YAML ooRexx Table objects, sequences to ooRexx arrays
- Ability to encode to YAML XML and decode from YAML XML
  - Uses either yaml.dtd (default) or yaml.xsd
    - JSON XML encoded files can be transformed to JSON with the supplied xmlToJson.xsl
  - New methods yamlToXml, yamlToXmlFile, parseXml, parseXmlFile

New YAML Class, Example "01_yaml.rex"

Read YAML from ooRexx Directive, Create Corresponding YAML File

```
-- 01_yaml.rex
y = .yaml~new
arr = .resources~yaml_data
say "resource yaml_data:"
say arr
say
obj = y~parseArray(arr)
say 'obj["title"]='obj["title"] '| version:' obj["version"]
say 'obj["authors"]='obj["authors"]~toString('L',',')
say
fn = "01_yaml_data.yaml"
say "saving to:" fn
.yaml~toYamlFile(obj, fn)
say
say "content of" fn":"
say .File~readChars(fn)

::resource yaml_data    -- YAML data sample
title: My Application
version: 2
authors:
   - Alice
   - Bob
::END

::requires "yaml.cls"   -- get YAML support
```

Output:

```
resource yaml_data:
title: My Application
version: 2
authors:
   - Alice
   - Bob

obj["title"]=My Application | version: 2
obj["authors"]=Alice,Bob

saving to: 01_yaml_data.yaml

content of 01_yaml_data.yaml:
authors: [Alice, Bob]
title: My Application
version: 2
```

New YAML Class, Example "02_yaml.rex"

Read YAML from ooRexx Directive, Create YAML XML File

```
-- 02_yaml.rex
y = .yaml~new
arr = .resources~yaml_data
say "resource yaml_data:"
say arr
say
obj = y~parseArray(arr)
fn = "02_yaml_data.xml"
say "create YAML-XML file:" fn
.yaml~yamlToXmlFile(obj, fn)
say
say "content of" fn":"
say .File~readChars(fn)

::resource yaml_data     -- YAML data sample
title: My Application
version: 2
authors:
   - Alice
   - Bob
::END

::requires "yaml.cls"    -- get YAML support
```

Output:

```
resource yaml_data:
title: My Application
version: 2
authors:
   - Alice
   - Bob

create YAML-XML file: 02_yaml_data.xml

content of 02_yaml_data.xml:
<?xml version="1.0" encoding="UTF-8"?>
<yaml xmlns="urn:yaml:xml:1.0" version="1.2">
  <document>
    <mapping>
      <entry>
        <key>
          <scalar type="str">authors</scalar>
        </key>
        <value>
          <sequence>
            <item>
              <scalar type="str">Alice</scalar>
            </item>
            <item>
              <scalar type="str">Bob</scalar>
            </item>
          </sequence>
        </value>
      </entry>

… continued …
```

```
… continued …

      <entry>
        <key>
          <scalar type="str">title</scalar>
        </key>
        <value>
          <scalar type="str">My Application</scalar>
        </value>
      </entry>
      <entry>
        <key>
          <scalar type="str">version</scalar>
        </key>
        <value>
          <scalar type="int">2</scalar>
        </value>
      </entry>
    </mapping>
  </document>
</yaml>
```


Prof. Rony G. Flatscher

Further Noteworthy Changes, 1

- New built-in-function (BIF) gc()
  - Invokes the ooRexx garbage collector if running the debug version of ooRexx
  - Meant for debugging native bridges between ooRexx and other environments as it causes pending UNINIT methods to be run which usually get used for maintaining reference counters
  - Please note: *never use it in production* as it may impact the performance if done wrongly, therefore let the interpreter execute it!
    - For that reason gc() by default gets executed in the debug version of ooRexx and returns .true, but will not execute a gc() call in the release version of ooRexx and returns .false in that case

Further Noteworthy Changes, 2

- Improved syntax error "13.1 – Incorrect character in program."
  - To display Unicode characters properly, ooRexx 5.2.0 starts to include and employ the quite popular Unicode library `"utf8proc"`
- This will allow to gradually introduce native Unicode support into ooRexx

Further Noteworthy Changes, 3

- Multithreading
  - Fixes a number of crashes and problems in native multithreading
  - No known multithreaded errors in ooRexx 5.2.0
  - Now `RexxInstance->Terminate()` may be safely invoked from any thread

Roundup

- Makes ooRexx fully dynamic in the area of global package options
  - Great for (temporarily) debugging programs/packages
- Improves support for JSON, allows XML renderings
- Introduces fully functional YAML support, allows XML renderings
- Introduces a `gc()`-BIF for debugging ooRexx bridges
- Incorporates `"utf8proc"` library for Unicode support
- Improves native multi-threading
- ***Use ooRexx 5.2.0*** *to replace older versions as soon as possible!*