



# NetRexx 4.05 GA Update

René Vincent Jansen

34th International Rexx Language Symposium, Almere, 20230515

# A new `-deps` option

- Intended usage is to assist in crafting build scripts that handle dependencies correctly
- First step is often to build an entire directory and list the dependencies
  - Sometimes impossible due to inter-language dependencies (Java, Groovy, Kotlin, Scala, etc)
- Lists the **exact** dependencies



```
) X NRC -NOVERBOSE -DEPS NSQL
CLASSNAME: NSQLBASELISTENER FILENAME: ./NSQLBASELISTENER.CLASS
UND CLASSNAME: NSQLBASELISTENER FILENAME: ./NSQLBASELISTENER.CLASS
E.TREE.PARSETREELISTENER DEP FOUND CLASSNAME: NSQLLISTENER FILENAME: ./NSQLLISTENER.
CLASSNAME: NETREXX.LANG.REXXDATE FILENAME: JAVA.UTIL.ZIP.ZIPFILE@1EE0005
CLASSNAME: NETREXX.LANG.REXXSTREAM FILENAME: JAVA.UTIL.ZIP.ZIPFILE@1EE0005
CLASSNAME: NETREXX.LANG.REXXREXX FILENAME: JAVA.UTIL.ZIP.ZIPFILE@1EE0005
CLASSNAME: JAVA.LANG.RUNNABLE FILENAME: /MODULES/JAVA.BASE/JAVA/LANG/RUNNABLE.CLASS
CLASSNAME: NETREXX.LANG.REXXDATE FILENAME: JAVA.UTIL.ZIP.ZIPFILE@1EE0005
CLASSNAME: NETREXX.LANG.REXXSTREAM FILENAME: JAVA.UTIL.ZIP.ZIPFILE@1EE0005
CLASSNAME: NETREXX.LANG.REXXREXX FILENAME: JAVA.UTIL.ZIP.ZIPFILE@1EE0005
E DEP FOUND CLASSNAME: JAVA.LANG.THROWABLE FILENAME: /MODULES/JAVA.BASE/JAVA/LANG/TH
E DEP FOUND CLASSNAME: JAVA.LANG.STRING FILENAME: /MODULES/JAVA.BASE/JAVA/LANG/STRIN
E DEP FOUND CLASSNAME: JAVA.LANG.STRING FILENAME: /MODULES/JAVA.BASE/JAVA/LANG/STRIN
NCE DEP FOUND CLASSNAME: JAVA.LANG.STRING FILENAME: /MODULES/JAVA.BASE/JAVA/LANG/STR
CONSTABLE DEP FOUND CLASSNAME: JAVA.LANG.STRING FILENAME: /MODULES/JAVA.BASE/JAVA/LA
CONSTANTDESC DEP FOUND CLASSNAME: JAVA.LANG.STRING FILENAME: /MODULES/JAVA.BASE/JAVA
DEP FOUND CLASSNAME: JAVA.SQL.SQLEXCEPTION FILENAME: /MODULES/JAVA.SQL/JAVA/SQL/SQLE
TIMEIT
CLASSNAME: TIMEIT FILENAME: ./TIMEIT.CLASS
FILE
CLASSNAME: JAVA.IO.FILE FILENAME: /MODULES/JAVA.BASE/JAVA/IO/FILE.CLASS
FILE
CLASSNAME: JAVA.IO.FILE FILENAME: /MODULES/JAVA.BASE/JAVA/IO/FILE.CLASS
E DEP FOUND CLASSNAME: JAVA.IO.FILE FILENAME: /MODULES/JAVA.BASE/JAVA/IO/FILE.CLASS
E DEP FOUND CLASSNAME: JAVA.IO.FILE FILENAME: /MODULES/JAVA.BASE/JAVA/IO/FILE.CLASS
EP FOUND CLASSNAME: JAVA.IO.OUTPUTSTREAM FILENAME: /MODULES/JAVA.BASE/JAVA/IO/OUTPUT
EP FOUND CLASSNAME: JAVA.IO.OUTPUTSTREAM FILENAME: /MODULES/JAVA.BASE/JAVA/IO/OUTPUT
ABLE DEP FOUND CLASSNAME: JAVA.IO.CLOSEABLE FILENAME: /MODULES/JAVA.BASE/JAVA/IO/CLC
E DEP FOUND CLASSNAME: JAVA.IO.PRINTSTREAM FILENAME: /MODULES/JAVA.BASE/JAVA/IO/PRIN
EP FOUND CLASSNAME: JAVA.IO.PRINTSTREAM FILENAME: /MODULES/JAVA.BASE/JAVA/IO/PRINTS1
FILEINPUTSTREAM
CLASSNAME: JAVA.IO.FILEINPUTSTREAM FILENAME: /MODULES/JAVA.BASE/JAVA/IO/FILEINPUTSTP
EP FOUND CLASSNAME: JAVA.IO.INPUTSTREAM FILENAME: /MODULES/JAVA.BASE/JAVA/IO/INPUTS1
ANTLRINPUTSTREAM
CLASSNAME: ORG.ANTLR.V4.RUNTIME.ANTLRINPUTSTREAM FILENAME: JAVA.UTIL.ZIP.ZIPFILE@E73
DEP
EP
```

# A build.xml for building an Android NetRexx APK

- Contributed by Jason Martin, this is added to the distribution

# A new `-noimplicituses` option

- The addition of Classic Rexx Compatibility BIFs into the same namespace introduces the risk of clashes
- These BIFs are **implicitly added** with the **uses** statement
- option `-noimplicituses` turns this around, in the case your program used these names already (as was the case with **date()**, which has an equivalent in Java)

# Deprecated org.netrexx.address

- Its intended function is done better by ADDRESS now
- In order not to break existing programs, we have used the *deprecation* mechanism to flag it, and possibly inspire users to use ADDRESS instead of the services of this package.

# #! shebangs are now supported

- specially for interpreted programs
- on operating environments that support them, NetRexx can use them now
- for example: see next page



```
→ NETREXX GIT:(MASTER) X CAT MNULLABLE.NRX
#! /USR/BIN/ENV NR
NAAM = 'AAP NOOT'
MAAN = REXX NULL

DO
  SAY NAAM MAAN
CATCH NULLPOINTEREXCEPTION
  SAY 'THAT DOES NOT WORK, MAAN IS NULL'
END

NAAM = REXX OPTIONAL.OFNULLABLE('AAP NOOT').ORELSE('NULL')
MAAN = REXX OPTIONAL.OFNULLABLE(REXX NULL).ORELSE('NULL')

SAY NAAM MAAN

MAAN = 'NONO'

SAY NAAM MAAN

MAAN = NULL

DO
  SAY NAAM MAAN
CATCH NULLPOINTEREXCEPTION
  SAY 'THAT DOES NOT WORK, MAAN IS NULL'
END
→ NETREXX GIT:(MASTER) X ./MNULLABLE.NRX
THAT DOES NOT WORK, MAAN IS NULL
AAP NOOT NULL
AAP NOOT NONO
THAT DOES NOT WORK, MAAN IS NULL
```

# Shebang

also, an example of using the Optional type from NetRexx



# nrc without CLASSPATH present

- On popular demand, Marc added this function to the startup scripts
- It enables anyone, on first install, to compile a NetRexx program without a CLASSPATH environment variable present
  - when using packages external to NetRexx, or run your compiled class with the **java** command, you will still need a CLASSPATH
- but we see it as a small step in simplification
- It works by having the script in a fixed location relative to the NetRexx jar

# user() and directory() added

- these are available in other Rexx variants and are useful for many scripts

- like:

```
/* REXX FOR WRITING ALL PRESENTATIONS FOR A YEAR TO A .TEX FILE */
YEAR=DIRECTORY().SUBSTR(DIRECTORY().LASTPOS('/')+1)
SAY DATE() TIME() 'STARTING WRITEYEAR FOR' YEAR
LINEOUT('PRESENTATIONS.TEX','% PRESENTATIONS FOR 'YEAR,1)

-- GET THE LOCATION
I6STEM=''; I6STEM=I6STEM
I6STEM[0]=2
O6UTSTEM=''
I6STEM[1]='CONNECT REXXLA;'
I6STEM[2]='SELECT LOCATION, STARTDATE, ENDDATE, ISBN FROM EVENT WHERE YEAR ='YEAR';'
ADDRESS SYSTEM 'MYSQL' WITH -
  INPUT STEM I6STEM -
  OUTPUT STEM O6UTSTEM
```

# ~~e2x and e2d accept strings~~

- no, they were taken out at the last moment, before GA, because we want to have a better consensus in the ARB about what these exactly should do
- because in Unicode, 1 character  $\leftrightarrow$  1 byte anymore; different ways to express those
- NetRexx never had these
- will probably return in NetRexx 4.06

# random() is added

- the random() BIF returns a random number between 0 and 999

```
READY; SAY RANDOM() 0.138 s
236
READY; SAY RANDOM()*1000 0.016 s
565000
READY; SAY RANDOM() 0.012 s
771
READY; SAY RANDOM() 0.008 s
445
READY; █ 0.015 s
```



# Pipes runner now reads source from stdin

- Marc Remes adapted the startup script 'Pipe' to be able to read in the pipe specification from STDIN

# A new ALL stage

- Jeff added an ALL stage to pipelines which behaves according to the CMS Pipelines specification

```
WORKSPACE FOR NETREXX 4.05 BUILD 109-20230510-1749
COPYRIGHT (C) MARTIN LAFAIX 2000
COPYRIGHT (C) PARTS REXXLA 2019,2021
PIPELINES PROCESSOR LOADED
READY; PIPE HELP ALL
/** ALL -- SELECT LINES CONTAINING STRINGS (OR NOT)

>>--ALL--+-----+--| EXPRESSION |---><
    +--%DEBUG--+ (6)
    +--%DUMP--+ (7)
    +--%SEE----+ (8)

NOTES:
(1) "EXPRESSION" CONSISTS OF ONE OR MORE DELIMITEDSTRINGS SEPARATED
    BY LOGICAL ANDS, ORS, AND NOTS, AND GROUPED, IF NEEDED, BY PARENTHESES.
(2) "&" IS USED FOR AND.
(3) SINCE "|" IS THE DEFAULT STAGE SEPARATOR, "!" MAY BE USED FOR OR.
(4) SINCE NETREXX USES "(" AND ")" FOR OPTIONS -- WHICH ARE NOT USED IN THE ALL STAGE --
    "[" AND "]" MUST BE USED FOR PARENTHESES.
(5) CMS PIPELINES, HAVING ORIGINATED ON 3270 TERMINALS, USES
```

# A new DELAY stage

- Jeff added an ALL stage to pipelines which behaves according to the CMS Pipelines specification

```
READY; PIPE HELP DELAY
```

```
/** DELAY -- DELAY THE STREAM FOR A SPECIFIED TIME
```

```
>>--DELAY--+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----><
          |  +---+---+                               |  +---EACH-(2)--+  | (1)
          +---+-----+---+---NUMBERHR:NUMBERMIN:NUMBERSEC+---+-----+---+ (1)
                    +---NUMBERMIN:NUMBERSEC-----+  +---ONCE-(2)--+ (1)
                    +---NUMBERSEC-----+ (1)
```

# Help is near (for all stages)

- For all stages there is now help available
- This is an incredibly useful features, which saves many trips to the documentation (books or pdf)



# Fix: address SYSTEM resolved at runtime

- this was an unexpected surprise; noticed this during a presentation where I ran a program that uses ADDRESS that was compiled on Linux, on a Windows box
- It looked for a zsh shell which was not there (yet?)
- This convinced us to evaluate the shell to run at **runtime** instead of at compile time

# #33 fix interpreter handling constant properties

- Interpreting code with **properties constant** yielded an error
- fixed by removing 'final' in the attributes

# Date() now available by default

- and everywhere available in a case insensitive way
- but use `-noimplicituses` if it is in the way and you somehow prefer the java `Date()` type

# NRWS fixes

- does not leave a lot of WS\*.nrx files around anymore
- setting history=off avoids writing a history file



# Homedirectory speed issues fixed

- Annoying 'phenomena' in macOS and Windows home directories that slow NetRexx startup down
- These have been surgically removed (they are avoided) during NetRexx translator startup
- The recommendation still is to use a subdirectory for development and not the user's home directory itself.