NetRexx 3.08 GA

New function presentation
René Jansen, 2019 International Rexx Language Symposium
Hursley Park, UK

Agenda

□ Translator changes

□ Pipelines improvements

□ Workspace for NetRexx

Translator changes

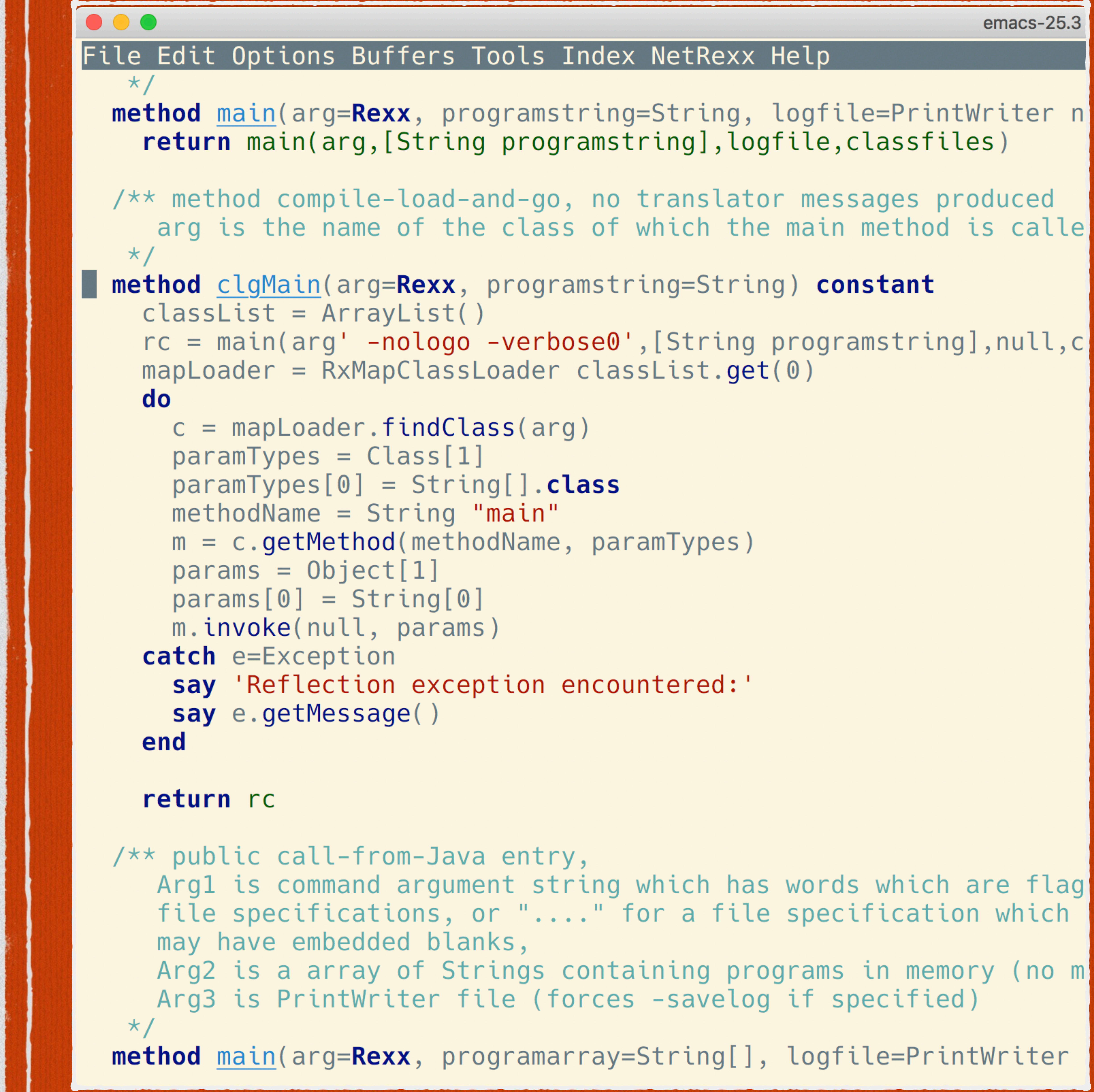
□ Save .class files with the right names

□ Move failed compiles to the bottom of the list (by Jason Martin)

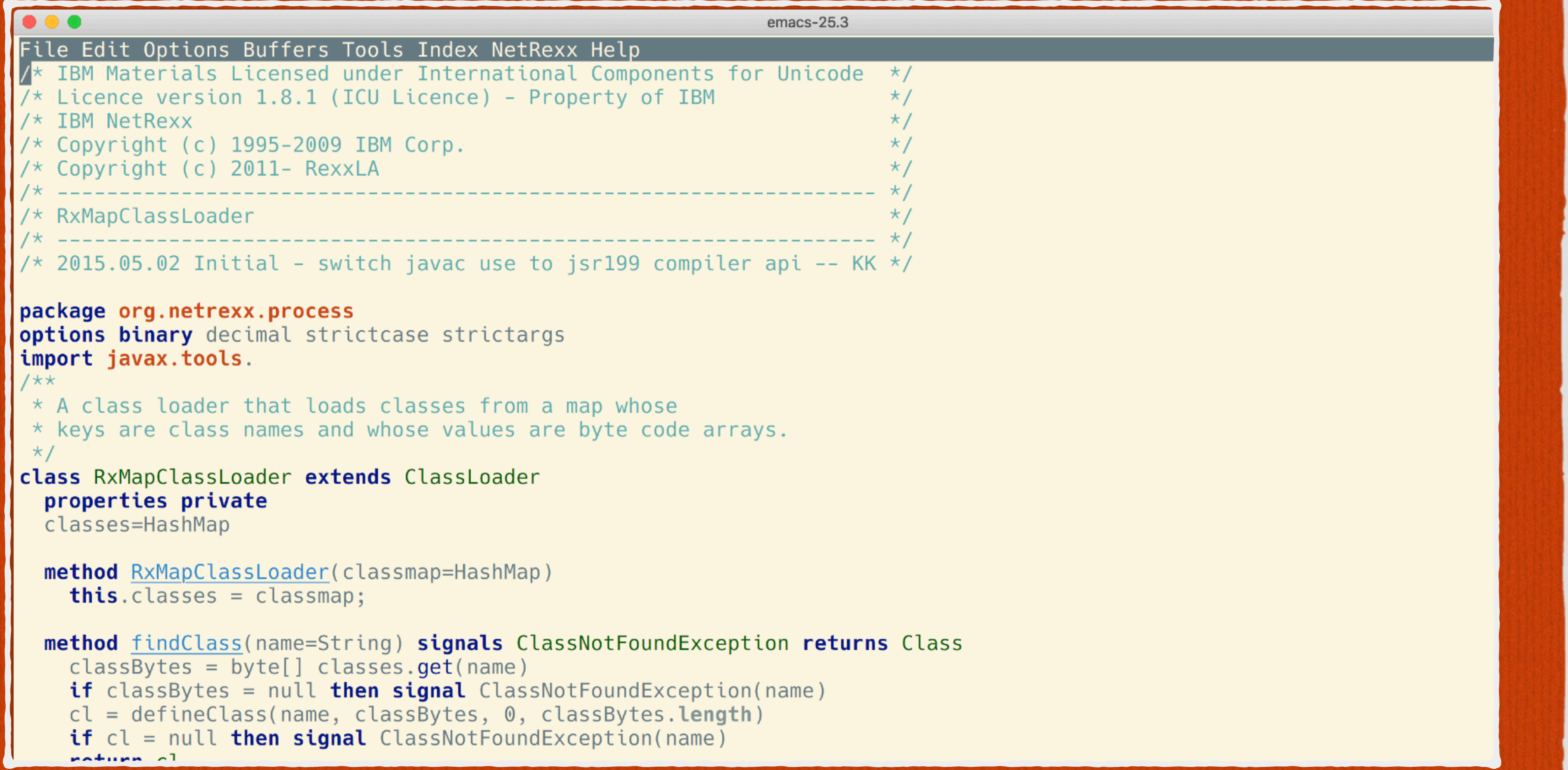
□ Suppress generics - related warnings

Translator Changes

- □ It works again on Java 6, and therefore on OS/2
- □ Added a NetRexxC.clgMain class for compile, load and go
  - □ This competes with the interpreter for development go-around time
  - □ Its main inspiration was the running of Pipelines in the shell



```
  */
method main(arg=Rexx, programstring=String, logfile=PrintWriter n
  return main(arg,[String programstring],logfile,classfiles)

/** method compile-load-and-go, no translator messages produced
   arg is the name of the class of which the main method is calle
  */
method clgMain(arg=Rexx, programstring=String) constant
  classList = ArrayList()
  rc = main(arg' -nologo -verbose0',[String programstring],null,c
  mapLoader = RxMapClassLoader classList.get(0)
  do
    c = mapLoader.findClass(arg)
    paramTypes = Class[1]
    paramTypes[0] = String[].class
    methodName = String "main"
    m = c.getMethod(methodName, paramTypes)
    params = Object[1]
    params[0] = String[0]
    m.invoke(null, params)
  catch e=Exception
    say 'Reflection exception encountered:'
    say e.getMessage()
  end

  return rc

/** public call-from-Java entry,
   Arg1 is command argument string which has words which are flag
   file specifications, or "...." for a file specification which
   may have embedded blanks,
   Arg2 is a array of Strings containing programs in memory (no m
   Arg3 is PrintWriter file (forces -savelog if specified)
  */
method main(arg=Rexx, programarray=String[], logfile=PrintWriter
```

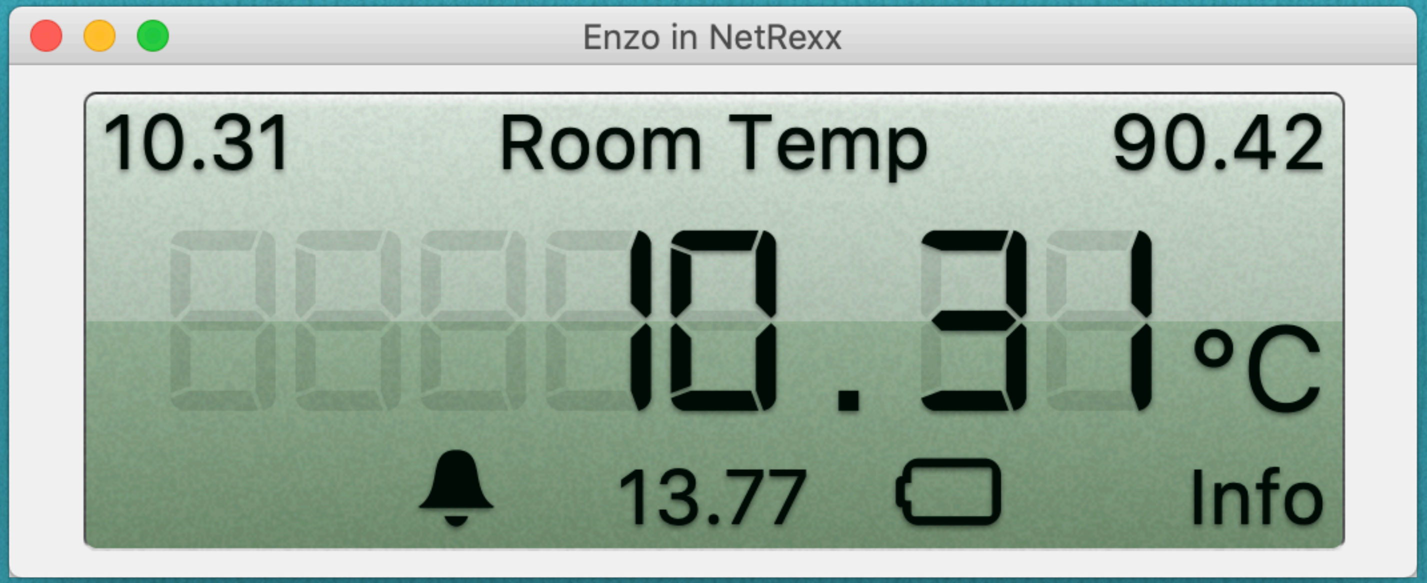


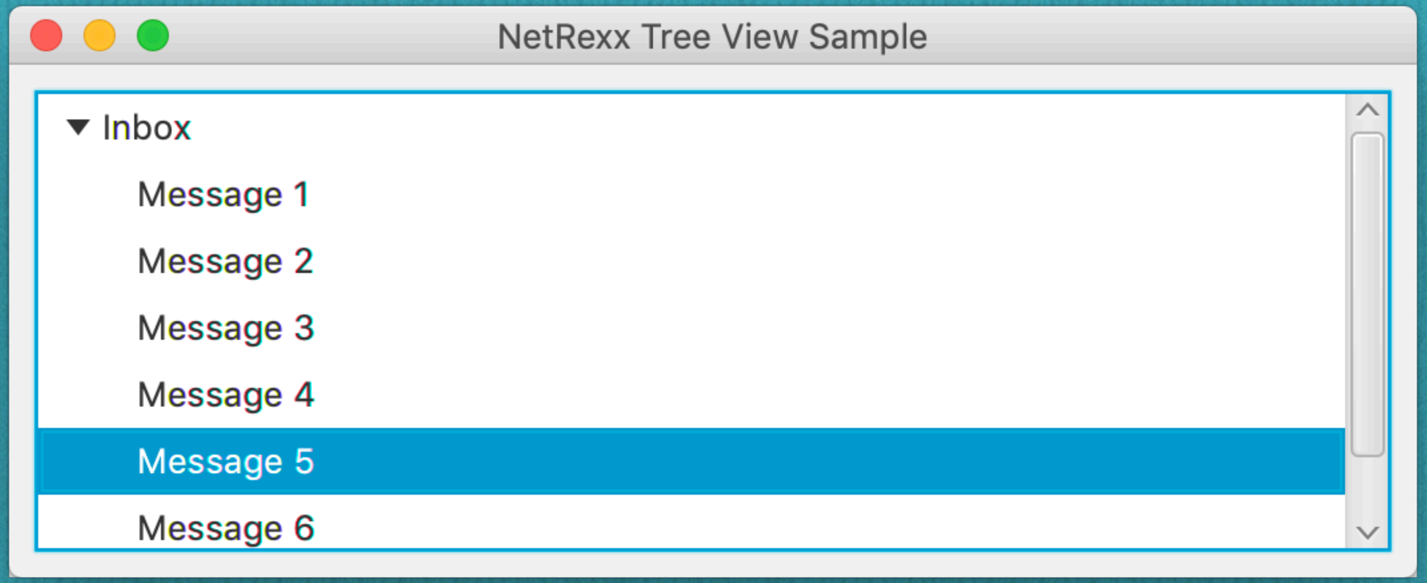
```
/* IBM Materials Licensed under International Components for Unicode  */
/* Licence version 1.8.1 (ICU Licence) - Property of IBM              */
/* IBM NetRexx                                                        */
/* Copyright (c) 1995-2009 IBM Corp.                                  */
/* Copyright (c) 2011- RexxLA                                         */
/* ------------------------------------------------------------------ */
/* RxMapClassLoader                                                   */
/* ------------------------------------------------------------------ */
/* 2015.05.02 Initial - switch javac use to jsr199 compiler api -- KK */

package org.netrexx.process
options binary decimal strictcase strictargs
import javax.tools.
/**
 * A class loader that loads classes from a map whose
 * keys are class names and whose values are byte code arrays.
 */
class RxMapClassLoader extends ClassLoader
  properties private
  classes=HashMap

  method RxMapClassLoader(classmap=HashMap)
    this.classes = classmap;

  method findClass(name=String) signals ClassNotFoundException returns Class
    classBytes = byte[] classes.get(name)
    if classBytes = null then signal ClassNotFoundException(name)
    cl = defineClass(name, classBytes, 0, classBytes.length)
    if cl = null then signal ClassNotFoundException(name)
    return cl
```

Added examples

□ The NetJoy language (Joy in NetRexx)

□ JavaFX compilation examples


Enzo in NetRexx
10.31
Room Temp
90.42
10.31 °C
13.77
Info


NetRexx Tree View Sample
Inbox
Message 1
Message 2
Message 3
Message 4
Message 5
Message 6

Pipelines

- This came about because I had a job last year working on z/VM and was struck by the perfect symbiosis of Rexx and Pipelines

- I remembered that we have Pipelines for NetRexx for some time, and showed this to VM people. The developments in this release are the result of these sessions

- In short, it should look more like the z/VM version and be faster

- I think these objectives are met

Pipelines

- Pipelines are now directly executable from the shell, instead of starting a compilation and an execution of the resulting class file
    - This uses the new NetRexxC.clgMain method
- Pipelines executed in the shell are now fast
- They need to be quoted
- Using the new (but in most parts 20 years old) Workspace for NetRexx, execution is fastest, and you don’t need the quotes. It’s like z/VM.

Pipelines

- New stages:
  - zip, unzip, ziplist, query [version | level]
  - Added <, >, and >> z/VM idioms for file input, output, append.
  - Take stage improved with less bugs and new secundary streams (Jeff Hennick)
  - Spec stage has a working C2X

Workspace for NetRexx

- □ By Martin Lafaix, 2000, with recent gracious permission to include
- □ Added Pipeline execution facility
- □ Added line editing on par with zsh (zsh is a unix shell) through JLine
- □ This also solves a change request for a more ‘RexxTry’ like interface to NetRexx than the nrxscript facility could offer

Workspace for NetRexx

- □ It remembers variable and method definitions
- □ It uses ‘frames’ so one can work on more sessions at a time
- □ Able to preload Pipelines machinery for instant execution