

CECIIS 2021

Employing Portable JavaFX GUIs with Scripting Languages

Rony G. Flatscher, Günter Müller

October 2021

WU

WIRTSCHAFTS
UNIVERSITÄT
WIEN VIENNA
UNIVERSITY OF
ECONOMICS
AND BUSINESS

Overview

- Some reasonings
- JavaFX
- Nutshell examples (ooRexx, Groovy, JRuby, Nashorn/JavaScript)
- Teaser
- Roundup
- Questions and answers
- Links

Some Reasonings, 1

- Scripting languages usually have no portable GUIs
- *Java*
 - Portable, including GUI classes!
 - *Java* scripting framework (*javax.script*, JSR-223)
 - Allows to turn any scripting language into a "Java scripting language"
 - Merely implement (*Abstract*)*ScriptEngine* and *ScriptEngineFactory*
- *Java* GUI classes
 - *awt* (abstract windows toolkit), *swing*
 - Rather involved, difficult for complex GUIs

Some Reasonings, 2

- *Java* GUI classes (continued)
 - *JavaFX*
 - Easy to create even complex GUIs interactively (*SceneBuilder*)
 - GUI definitions can be descriptively saved in *FXML* text files
 - Supports *javafx.script*, hence any *Java* scripting language
 - ➔ Makes *JavaFX* GUI objects available via the *ScriptContext*
- *ooRexx* scripting language examples have been developed for WU students
 - Demonstrate the *JavaFX* architecture and powerful abilities
 - *ooRexx* samples can be converted to any *Java* scripting language!

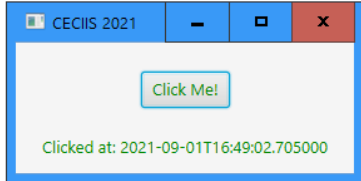
Background (ooRexx)

- Business administration students at WU, who are learning to program
 - *ooRexx*: easy syntax, dynamically typed, caseless, message based
 - Within a four hour lecture (for four months/single semester) they become empowered from zero to creating programs that exploit *MS Office*, *OpenOffice*, socket programming and *JavaFX* GUIs
 - *ooRexx-Java* bridge "*BSF4ooRexx*"
 - Implements *javax.script*, turns *ooRexx* into a *Java* scripting language
 - Includes support to camouflage *Java* as *ooRexx* (e.g. messages)
- Beginners become able to create even complex GUIs exploiting *JavaFX*

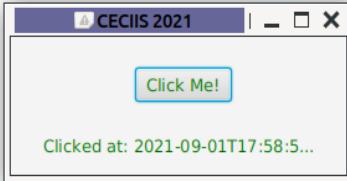
- Originally developed as a stand-alone GUI replacement for awt/swing
 - Originally included the scripting language "*JavaFX Script*", later removed
 - Hence support for *javafx.script* available
 - Any scripting language with *javafx.script* support can be deployed
 - Targeted for mobile and desktop applications
- *JavaFX* GUIs can be created either by
 - Directly instantiating and configuring the *JavaFX* GUI classes
 - Or using *SceneBuilder* to create an XML file that defines the *JavaFX* GUI
 - *JavaFX* class *FXMLLoader* will load, setup and instrumentate the GUI

Nutshell Example (ooRexx) Same GUI on Windows, Linux, MacOS

```
G:\tmp\ceciis2021\oorexx>main.rex  
REXXout>REXX-ooRexx_5.0.0(MT)_32-bit 6.05 13 Aug 2021
```



```
rony@rony-linux:~/Dropbox/xfer/temp/ceciis2021/oorexx$ rexxj.sh main.rex  
Gtk-Message: 17:58:52.031: Failed to load module "topmenu-gtk-module"  
REXXout>REXX-ooRexx_5.0.0(MT)_64-bit 6.05 29 Aug 2021
```



```
rony@ronymac2014 oorexx % rexxj.sh main.rex  
REXXout>REXX-ooRexx_5.0.0(MT)_64-bit 6.05 18 Jul 2021
```

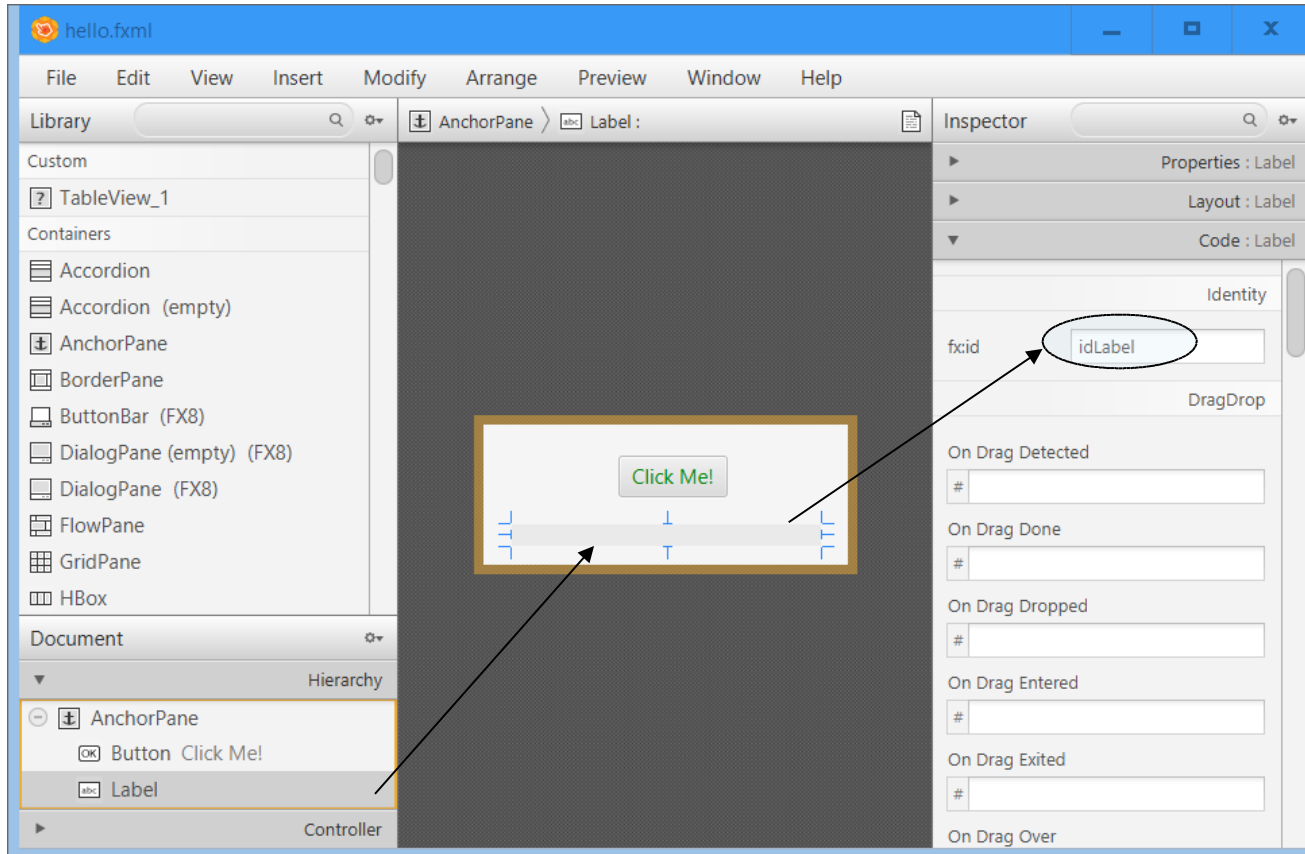


Note: **green** color!

Nutshell Example (ooRexx) FXML Definitions ("*hello.fxml*")

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?language rexx?>
<AnchorPane id="AnchorPane" prefHeight="104.0" prefWidth="270.0"
  xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <!-- JavaFX runs the ooRexx code in the 'onAction' attribute -->
    <Button fx:id="idButton" layoutX="100.0" layoutY="23.0"
      onAction="/* @get(idLabel) */; idLabel~text=buttonClicked()"
      text="Click Me!" textFill="GREEN" />
    <Label fx:id="idLabel" alignment="CENTER" contentDisplay="CENTER"
      layoutX="21.0" layoutY="74.0" minHeight="16" minWidth="49"
      prefHeight="16.0" prefWidth="229.0" textFill="GREEN" />
  </children>
  <!-- call REXX program, makes the routine "buttonClicked" visible -->
  <fx:script source="hello_controller.rex" />
</AnchorPane>
```


Nutshell Example (ooRexx) SceneBuilder (Editing "hello.fxml")



Nutshell Example (ooRexx)

"hello_controller.rex"

```
-- Controller routine defines public routine buttonClicked()  
parse version v; say v
```

```
::routine buttonClicked public  
return "Clicked at:" .dateTime~new
```

Nutshell Example (ooRexx)

"main.rex"

- Used for running all nutshell examples, i.e. ooRexx, Groovy, JRuby, Nashorn
 - Place in the respective subdirectories, loads and runs "hello.fxml"

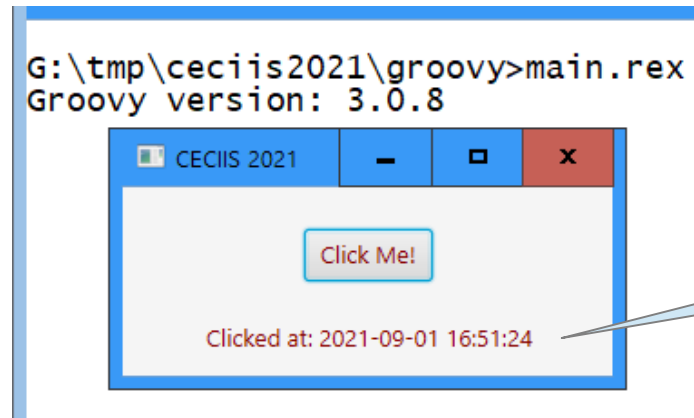
```
#!/usr/bin/env rexx
rxApp=.RexxApplication~new      -- Rexx class implements "start" method
jrxApp=BSFCreateRexxProxy(rxApp, "javafx.application.Application")
jrxApp~launch(jrxApp~getClass, .nil) -- launch JavaFX Application
call sysSleep 0.1              -- sleep a bit

::requires "BSF.CLS"           -- get the ooRexx-Java bridge

/* Rexx class: implements method "start" of "javafx.application.Application" */
::class RexxApplication        -- defines a Rexx class

::method start                 -- implements abstract method "start"
  use arg primaryStage         -- fetch the primary stage (window)
  primaryStage~title="CECIIS 2021" -- set stage (window) title
  -- create a URL object for the file "hello.fxml"
  fxmlUrl=.bsf~new("java.net.URL", "file:hello.fxml")
  -- load the FXMLLoader class, load the FXML file, returns DOM's root
  rootNode=bsf.loadClass("javafx.fxml.FXMLLoader")~load(fxmlUrl)
  scene=.bsf~new("javafx.scene.Scene", rootNode) -- create the scene
  primaryStage~setScene(scene) -- set the stage to our scene
  primaryStage~show           -- show the stage
```

JavaFX with Groovy



Note: **maroon** color!

Nutshell Example (Groovy)

FXML Definitions ("*hello.fxml*")

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?language groovy?>
<AnchorPane id="AnchorPane" prefHeight="104.0" prefWidth="270.0"
  xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <Button fx:id="idButton" layoutX="100.0" layoutY="23.0"
      onAction="idLabel.setText(buttonClicked())"
      text="Click Me!" textFill="MAROON" />
    <Label fx:id="idLabel" alignment="CENTER" contentDisplay="CENTER"
      layoutX="21.0" layoutY="74.0" minHeight="16" minWidth="49"
      prefHeight="16.0" prefWidth="229.0" textFill="MAROON" />
  </children>
  <fx:script source="hello_controller.groovy" />
</AnchorPane>
```

Nutshell Example (Groovy)

"hello_controller.groovy"

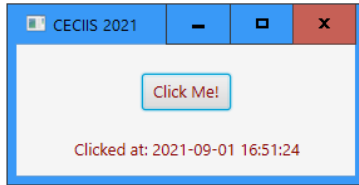
```
// Controller routine buttonClicked() in Groovy  
println "Groovy version: " + GroovySystem.version
```

```
def buttonClicked () {  
    def now = new java.util.Date()  
    def df = new java.text.SimpleDateFormat("yyyy-MM-dd HH:mm:ss")  
    return "Clicked at: " + df.format(now);  
}
```

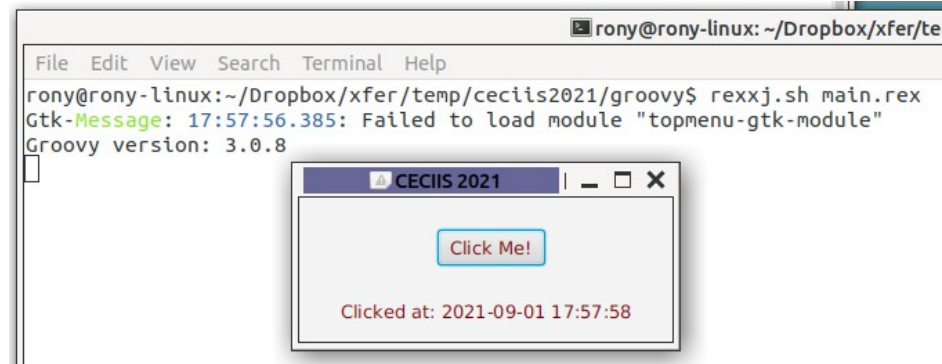
Nutshell Example (Groovy)

Same GUI on Windows, Linux, MacOS

```
G:\tmp\ceciis2021\groovy>main.rex  
Groovy version: 3.0.8
```

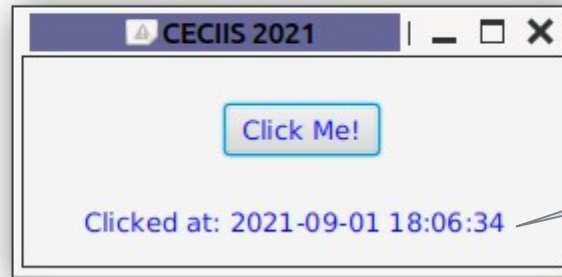


```
rony@ronymac2014 groovy % rexxj.sh main.rex  
Groovy version: 3.0.8
```



JavaFX with JRuby

```
rony@rony-linux:~/Dropbox/xfer/temp/ceciis2021/jruby$ rexxj.sh main.rex  
Gtk-Message: 18:05:43.531: Failed to load module "topmenu-gtk-module"  
JRUBY_VERSION: 9.2.19.0 RUBY_VERSION: 2.5.8
```



Note: **blue** color!

Nutshell Example (JRuby) FXML Definitions ("*hello.fxml*")

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?language jruby?>
<AnchorPane id="AnchorPane" prefHeight="104.0" prefWidth="270.0"
  xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <Button fx:id="idButton" layoutX="100.0" layoutY="23.0"
      onAction="idLabel.setText buttonClicked()"
      text="Click Me!" textFill="BLUE" />
    <Label fx:id="idLabel" alignment="CENTER" contentDisplay="CENTER"
      layoutX="21.0" layoutY="74.0" minHeight="16" minWidth="49"
      prefHeight="16.0" prefWidth="229.0" textFill="BLUE" />
  </children>
  <fx:script source="hello_controller.rb" />
</AnchorPane>
```

Nutshell Example (JRuby)

"hello_controller.rb"

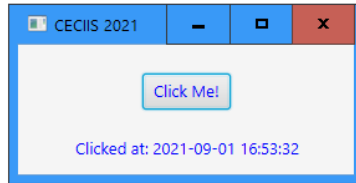
```
# Controller routine buttonClicked() in JRuby
puts "JRUBY_VERSION: " + JRUBY_VERSION + " RUBY_VERSION: " + RUBY_VERSION

def buttonClicked ()
  return "Clicked at: " + Time.new.strftime("%Y-%m-%d %H:%M:%S")
end
```

Nutshell Example (JRuby)

Same GUI on Windows, Linux, MacOS

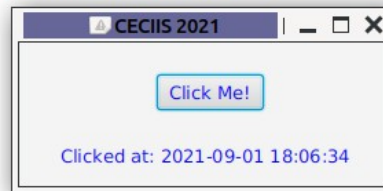
```
G:\tmp\cecii2021\jruby>main.rex  
JRUBY_VERSION: 9.2.19.0 RUBY_VERSION: 2.5.8
```



```
[rony@ronymac2014 jruby % rexxj.sh main.rex  
JRUBY_VERSION: 9.2.19.0 RUBY_VERSION: 2.5.8
```



```
rony@rony-linux:~/Dropbox/xfer/temp/cecii2021/jruby$ rexxj.sh main.rex  
Gtk-Message: 18:05:43.531: Failed to load module "topmenu-gtk-module"  
JRUBY_VERSION: 9.2.19.0 RUBY_VERSION: 2.5.8
```



JavaFX with Nashorn/JavaScript

```
G:\tmp\ceciis2021\nashorn>main.rer  
engine name: Oracle Nashorn, language version: ECMA - 262 Edition 5.1
```



Note: **magenta** color!

Nutshell Example (Nashorn/JavaScript) FXML Definitions ("*hello.fxml*")

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?language nashorn?>
<AnchorPane id="AnchorPane" prefHeight="104.0" prefWidth="270.0"
  xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <Button fx:id="idButton" layoutX="160.0" layoutY="23.0"
      onAction="idLabel.setText( buttonClicked() );"
      text="Click Me!" textFill="MAGENTA" />
    <Label fx:id="idLabel" alignment="CENTER" contentDisplay="CENTER"
      layoutX="21.0" layoutY="74.0" minHeight="16" minWidth="49"
      prefHeight="16.0" prefWidth="389.0" textFill="MAGENTA" />
  </children>
  <fx:script source="hello_controller.js" />
</AnchorPane>
```

Nutshell Example (Nashorn/JavaScript)

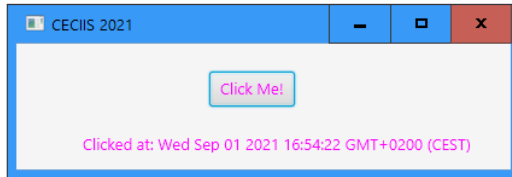
"hello_controller.js"

```
// Controller routine buttonClicked() in Nashorn
var factory = (new (Java.type("javax.script.ScriptEngineManager"))).
    getEngineByName("nashorn").getFactory() ;
print("engine name: "      +factory.getEngineName()+
      ", language version: "+factory.getLanguageVersion());

function buttonClicked() {
    return "Clicked at: " + new Date() ;
}
```

Nutshell Example (Nashorn/JavaScript) Same GUI on Windows, Linux, MacOS

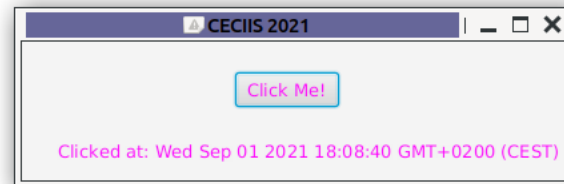
```
G:\tmp\ceciis2021\nashorn>main.rex  
engine name: Oracle Nashorn, language version: ECMA - 262 Edition 5.1
```



```
[rony@ronymac2014 nashorn % rexxj.sh main.rex  
engine name: Oracle Nashorn, language version: ECMA - 262 Edition 5.1  
]
```



```
rony@rony-linux:~/Dropbox/xfer/temp/ceciis2021/nashorn$ rexxj.sh main.rex  
Gtk-Message: 18:08:00.666: Failed to load module "topmenu-gtk-module"  
Warning: Nashorn engine is planned to be removed from a future JDK release  
Warning: Nashorn engine is planned to be removed from a future JDK release  
engine name: Oracle Nashorn, language version: ECMA - 262 Edition 5.1  
]
```



Teaser

"bsf4oorex/samples/JavaFX/fxml_07/MortgageCalculator.rex"

- Could be *easily* rewritten to Groovy, JRuby, Nashorn/JavaScript, ...

Credit Amount	250000.00
	€ 250.000,00
Interest Rate	4.7
# Months	206
Monthly Payment Rate	€ 1.770,56
Accrued Payments	€ 364.735,07

Calculate

- As demonstrated *JavaFX* GUIs can be used from scripting languages
 - Prerequisite: an implementation of *javax.script.ScriptEngine[Factory]*
- Scripting languages can use and control even the most complex GUIs
 - *SceneBuilder* allows *JavaFX* GUIs to be created interactively
 - Resulting *FXML* files can instrumentate (multiple) scripting languages
- The *JavaFX* GUIs are platform independent, hence
 - Multiplatform scripts can exploit portable *JavaFX* GUIs
- *Hint*: make sure to download *Java/OpenJDK* with *JavaFX* contained

Questions ?

Links (as of 2021-10-08), 1

- OpenJDK (make sure JavaFX is contained!), e.g.:
 - „jdk-fx“: <<https://www.azul.com/downloads/?package=jdk-fx>>
 - „Full JDK“: <<https://bell-sw.com/pages/downloads/>>
- SceneBuilder: <<https://gluonhq.com/products/scene-builder/>>
- ooRexx 5.0: <<https://sourceforge.net/projects/ooress/files/ooress/5.0.0beta/>>
- BSF4ooRexx (ooRexx-Java bridge): <<https://sourceforge.net/projects/bsf4ooress/files/beta/20200928/>>
 - Note: there are quite a few JavaFX nutshell examples available that demonstrate what can be done with JavaFX and that could be ported to other scripting languages; cf. "samples/JavaFX/index.html" in the BSF4ooRexx installation directory
- Groovy: <<https://groovy.apache.org/download.html>>
- JRuby: <<https://www.jruby.org/download>>
- Nashorn module for OpenJDK 15 and later: <<https://github.com/openjdk/nashorn>>

Links (as of 2021-10-08), 2

Selected links from article:

- *RexxScript* – *Rexx* Scripts Hosted and Evaluated by *Java* (Package *javax.script*):
<<https://www.rexxla.info/events/2017/presentations/201704-RexxScript-Article.pdf>>
- *JavaFX* for *ooRexx*: <<https://www.rexxla.info/events/2017/presentations/201711-ooRexx-JavaFX-Article.pdf>>
- Anatomy of a GUI (Graphical User Interface):
<<https://www.rexxla.info/events/2018/presentations/201803-AnatomyOfGUI-Article.pdf>>,
<<https://epub.wu.ac.at/6875/>>



VIENNA UNIVERSITY OF
ECONOMICS AND BUSINESS

Prof. Mag. Dr. Rony G. Flatscher
Welthandelsplatz 1
1020 Wien/Vienna

Austria/Europe

Mail: rony.flatscher@wu.ac.at
Phone: +43-1-31336-4881