

# The IntelliJ IDEA Plugin ooRexxPlugin 2.2 for Rexx and ooRexx 5.0

The 2023 International Rexx Symposium  
Almere, The Netherlands  
May 14<sup>th</sup> – May 17<sup>th</sup> 2023

# Overview

---

- IntelliJ Plugin
- ooRexx Plugin 2.2
- ooRexxDoc
- Roundup
- Links

# IntelliJ (by JetBrains)

- IDEA
  - Integrated Development Environment Application
  - Professional software development environment for e.g.
    - C++, Groovy, Java, JavaScript, Kotlin, PHP, Python, Ruby, Rust, Scala, ...
- Available for all major operating systems
  - Apple, Linux, Windows
- Free community edition
  - Professional edition free for academia (faculty, students)
- Open plugin architecture
- Download URL (as of 2023-05-11)  
<https://www.jetbrains.com/idea/download>

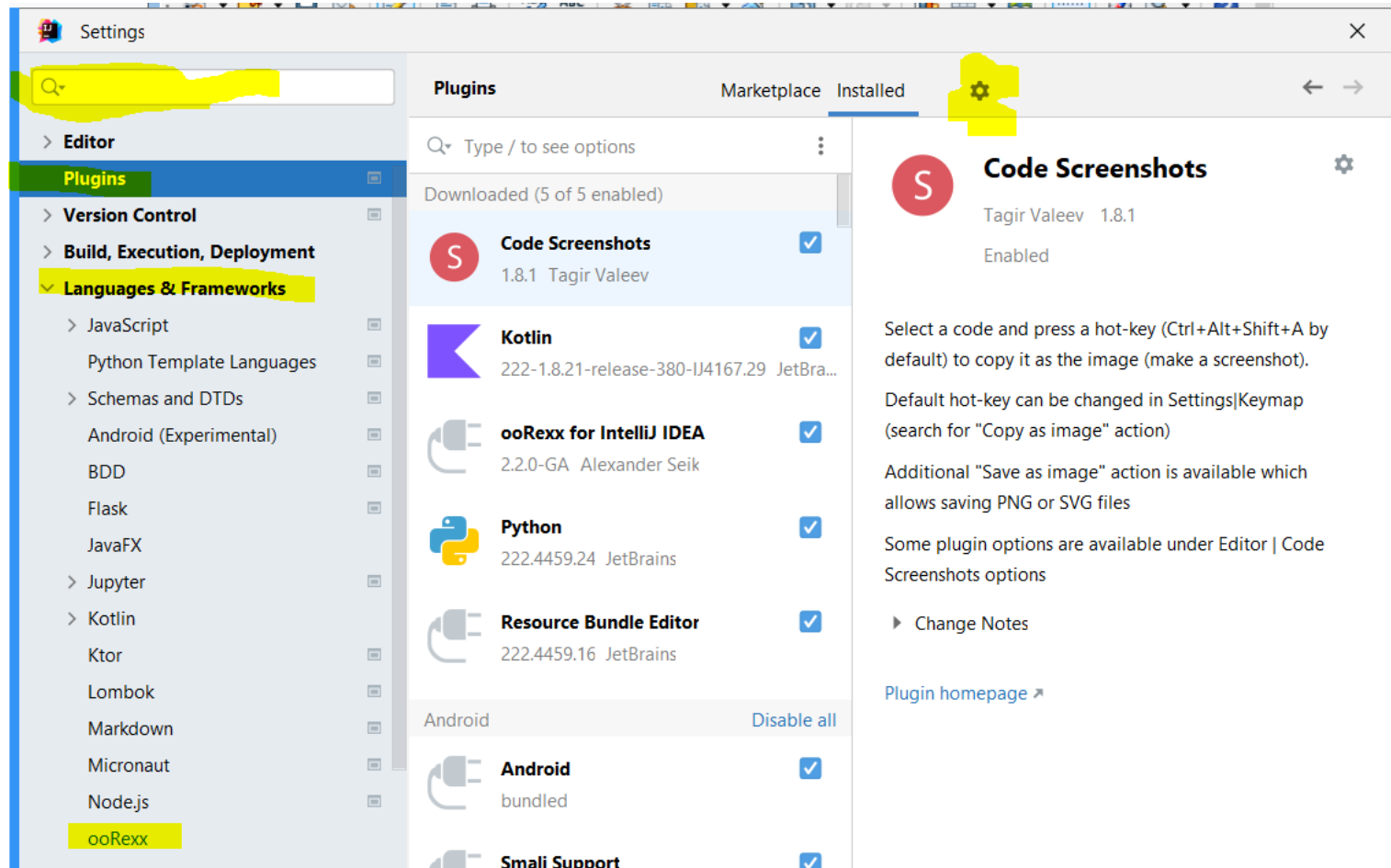
# ooRexxPlugin 2.2

- IntelliJ plugin by Alexander Seik
- Introduced at Rexx Symposium 2017
- Adds full ooRexx support to IntelliJ
  - Syntax checking, syntax highlighting, folding, ...
  - Allows for on-the fly ooRexxDoc creation (right mouse click on program to doc)
    - ooRexxDoc downloadable as a separate, standalone tool
  - Introduces auto completion!
  - Supports mainframe Rexx and Jean-Louis Faucher's "executor"
- Download URL and [README.txt](#)  
<https://sourceforge.net/projects/bsf4oorexx/files/Sandbox/aseik/ooRexxIDEA/GA/2.2.0/>

# ooRexxPlugin 2.2 Installation

- Download [ooRexxPlugin-2.2.0-GA.zip](#)
- IntelliJ → File → Settings ... → Plugins → Gear Icon → Click
  - Choose download directory and select [ooRexxPlugin-2.2.0-GA.zip](#) → Enter
  - Restart IntelliJ IDEA to make the ooRexx plugin active

# ooRexxPlugin 2.2 Installation (Windows Screenshot)



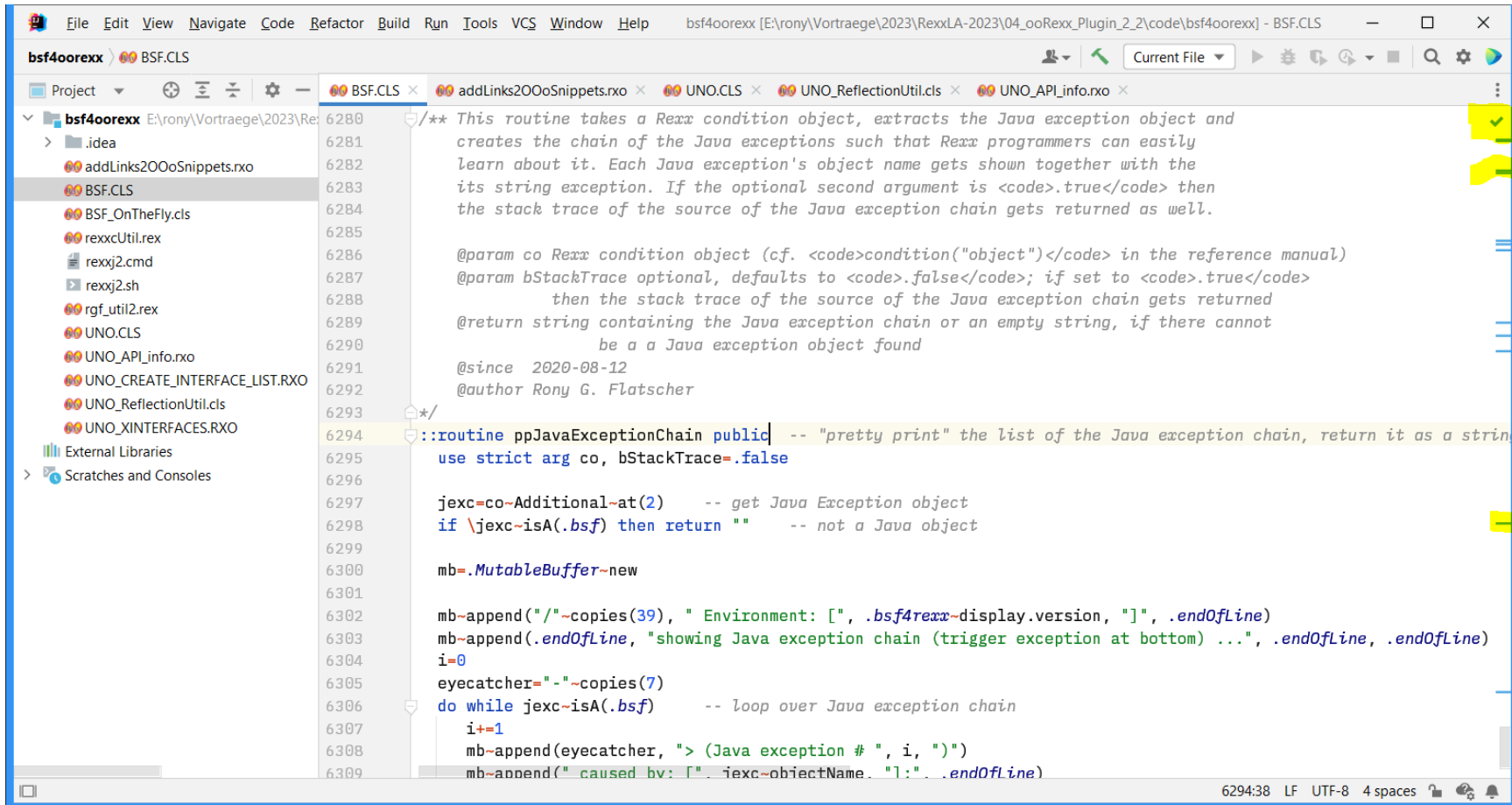
# ooRexxPlugin 2.2 Configuration

- IntelliJ → File → Settings ...
  - Enter ooRexx in the search field on the upper left hand side

Check for mainframe Rexx & executor

The screenshot shows the IntelliJ Settings dialog box. The search field at the top left contains 'ooRexx'. The left sidebar shows the 'Languages & Frameworks' section expanded, with 'ooRexx' selected. The main panel displays the 'ooRexx' configuration options under 'Languages & Frameworks'. The 'Allow special characters (@, #, €, \$). Requires restart.' checkbox is checked. The 'Enable executor parsing (https://github.com/jlfaucher/executor)' checkbox is unchecked. The 'Auto Completion Casing. Requires restart.' dropdown menu is set to 'Camel Case'. A blue callout bubble points to the 'Enable executor parsing' checkbox with the text 'Check for mainframe Rexx & executor'.

# IntelliJ with oRexxPlugin: BSF.CLS, 1



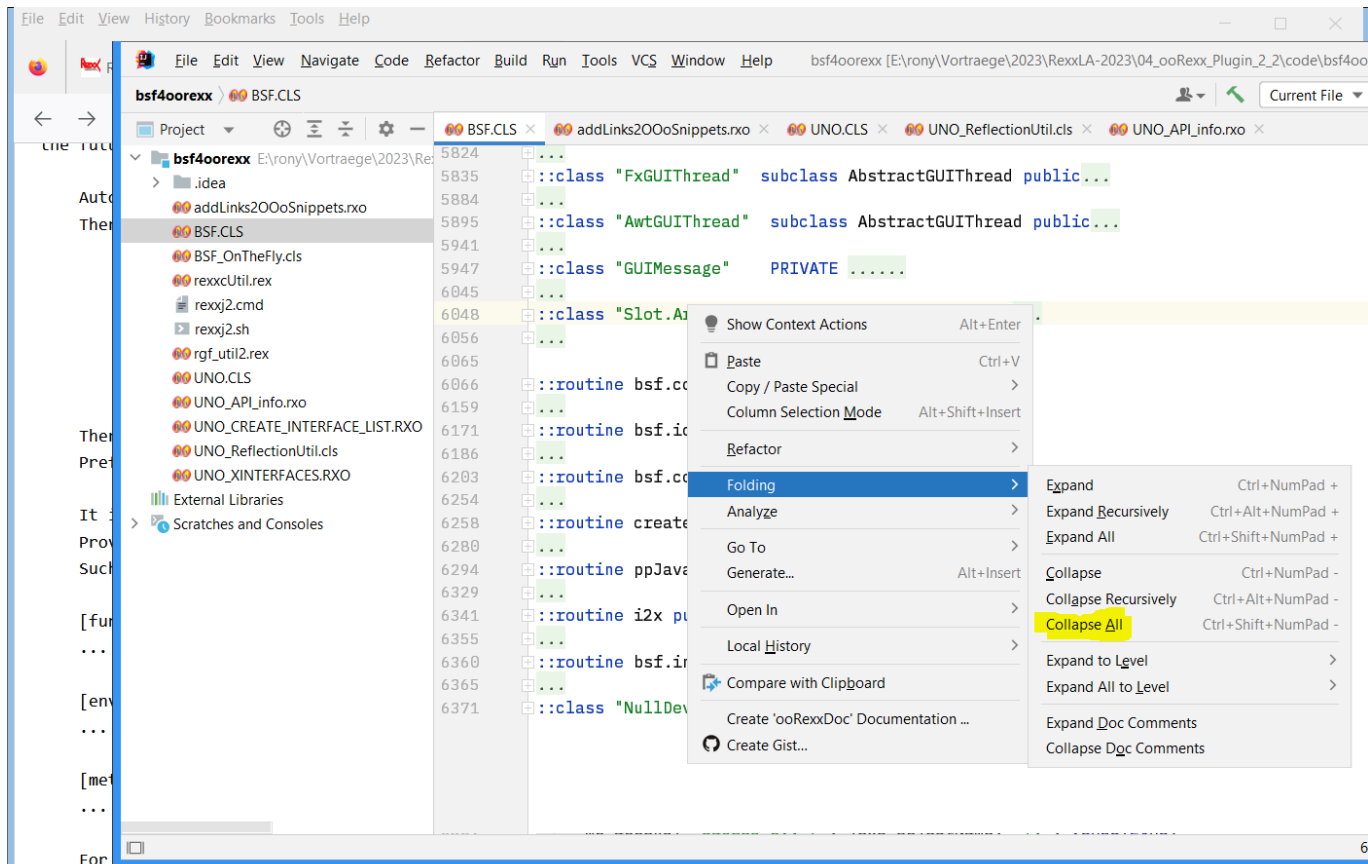
```
6280 /** This routine takes a Rexx condition object, extracts the Java exception object and
6281 creates the chain of the Java exceptions such that Rexx programmers can easily
6282 learn about it. Each Java exception's object name gets shown together with the
6283 its string exception. If the optional second argument is <code>.true</code> then
6284 the stack trace of the source of the Java exception chain gets returned as well.
6285
6286 @param co Rexx condition object (cf. <code>condition("object")</code> in the reference manual)
6287 @param bStackTrace optional, defaults to <code>.false</code>; if set to <code>.true</code>
6288 then the stack trace of the source of the Java exception chain gets returned
6289 @return string containing the Java exception chain or an empty string, if there cannot
6290 be a a Java exception object found
6291 @since 2020-08-12
6292 @author Rony G. Flatscher
6293
6294 */
6295 ::routine ppJavaExceptionChain public | -- "pretty print" the list of the Java exception chain, return it as a string
6296 use strict arg co, bStackTrace=.false
6297
6298 jexc=co-Additional-at(2) -- get Java Exception object
6299 if \jexc-isA(.bsf) then return "" -- not a Java object
6300
6301 mb=.MutableBuffer-new
6302
6303 mb-append("/"-copies(39), " Environment: [", .bsf4rex-Display.version, "]", .endOfLine)
6304 mb-append(.endOfLine, "showing Java exception chain (trigger exception at bottom) ...", .endOfLine, .endOfLine)
6305 i=0
6306 eyecatcher="-"-copies(7)
6307 do while jexc-isA(.bsf) -- loop over Java exception chain
6308 i+=1
6309 mb-append(eyecatcher, "> (Java exception # ", i, ")")
6310 mb-append(" caused by: [" . jexc-objectName . "]: " . jexc-exceptionText . .endOfLine)
```

No errors!

TODO-items



# IntelliJ with ooRexxPlugin: **BSF.CLS**, 2 (Collapsed)

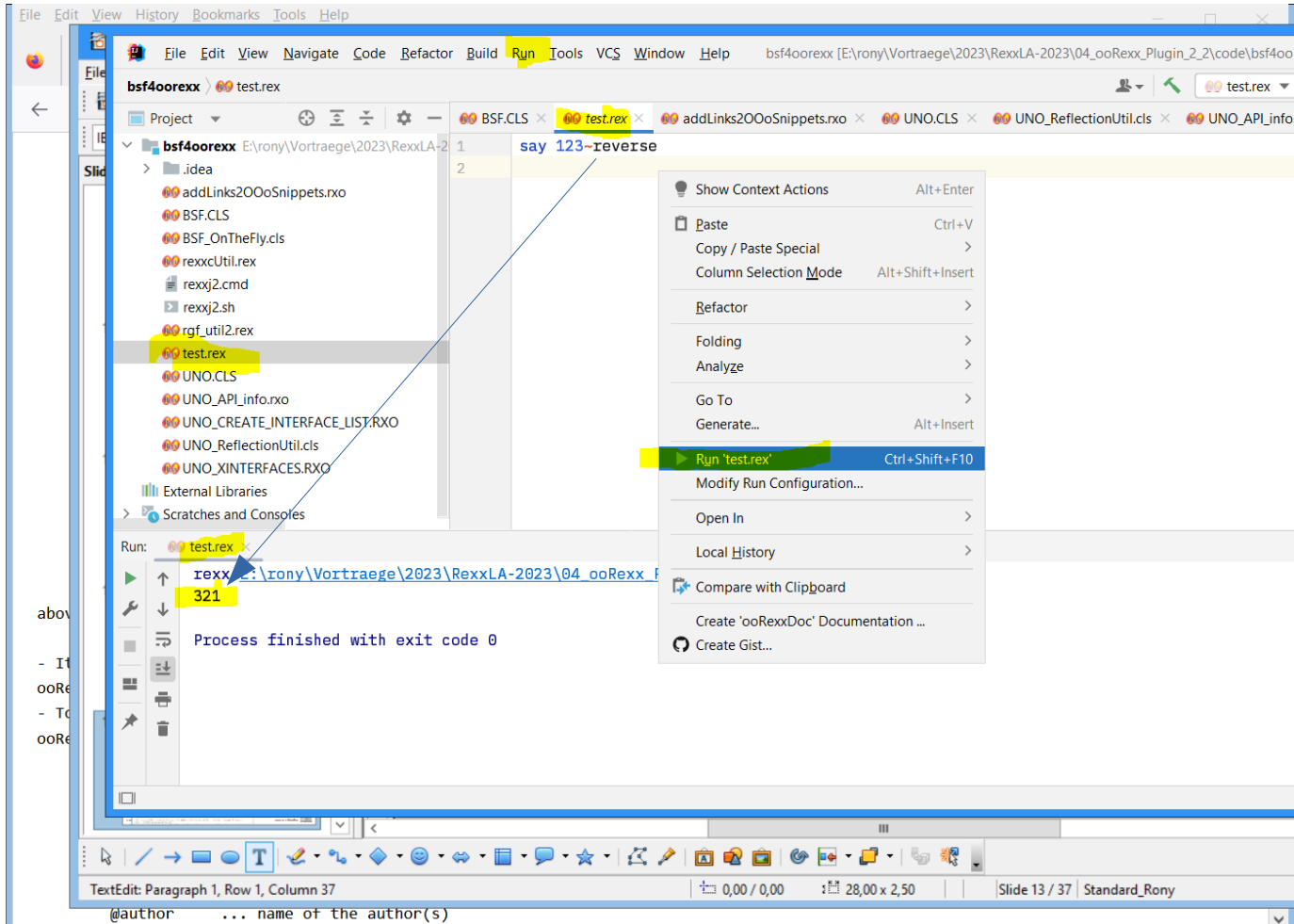


above.

9

- It is also possible to change the text mode of the plugin to Mainframe-Rexx via File -> Settings -> Languages & Frameworks -> ooRexx -> checkbox 'allow special characters'

# IntelliJ with oRexxPlugin: Running from IDEA



# ooRexxPlugin 2.2, Auto Completion, 1

- Three categories of auto completion names
  - Public functions (routines)
  - Environment symbols
  - Methods
- Possible to extend auto completion list
  - Create the subdirectory `IntelliJ/ooRexxPlugin` in your home directory
    - `%USERPROFILE%\IntelliJ\ooRexxPlugin` (Windows) or `"$HOME/IntelliJ/ooRexxPlugin"` (Unix)
  - Text file with list of names organized in sections with the header lines
    - `"[functions]", "[environment symbols]", "[methods]"`
    - Copy text file to `"IntelliJ/ooRexxPlugin"` in your home directory
    - Restart IntelliJ

# ooRexxPlugin 2.2, Auto Completion, 2

- Simple program to create such name completion files  
`extractNamesFromRexxPackages.rex < program.rex > program.rex.txt`
  - Copy `program.rex.txt` to `IntelliJ/ooRexxPlugin` in your home directory
  - Restart IntelliJ
- Zip archive `rgfNames.zip`
  - `IntelliJ/ooRexxPlugin/BSF.CLS.txt`
  - `IntelliJ/ooRexxPlugin/CLR.CLS.txt`
  - `IntelliJ/ooRexxPlugin/rgf_util2.rex.txt`
  - `IntelliJ/ooRexxPlugin/UNO.CLS.txt`
  - `IntelliJ/ooRexxPlugin/dbus.cls.txt`

# ooRexxPlugin 2.2, Auto Completion, 3

The screenshot shows an IDE window titled 'bsf4ooress' with a project named 'test.rexx'. The main editor displays the following code:

```
1 say 123-reverse  
2 a=someObject-run
```

An auto-completion popup menu is visible over the 'run' method call, listing the following options:

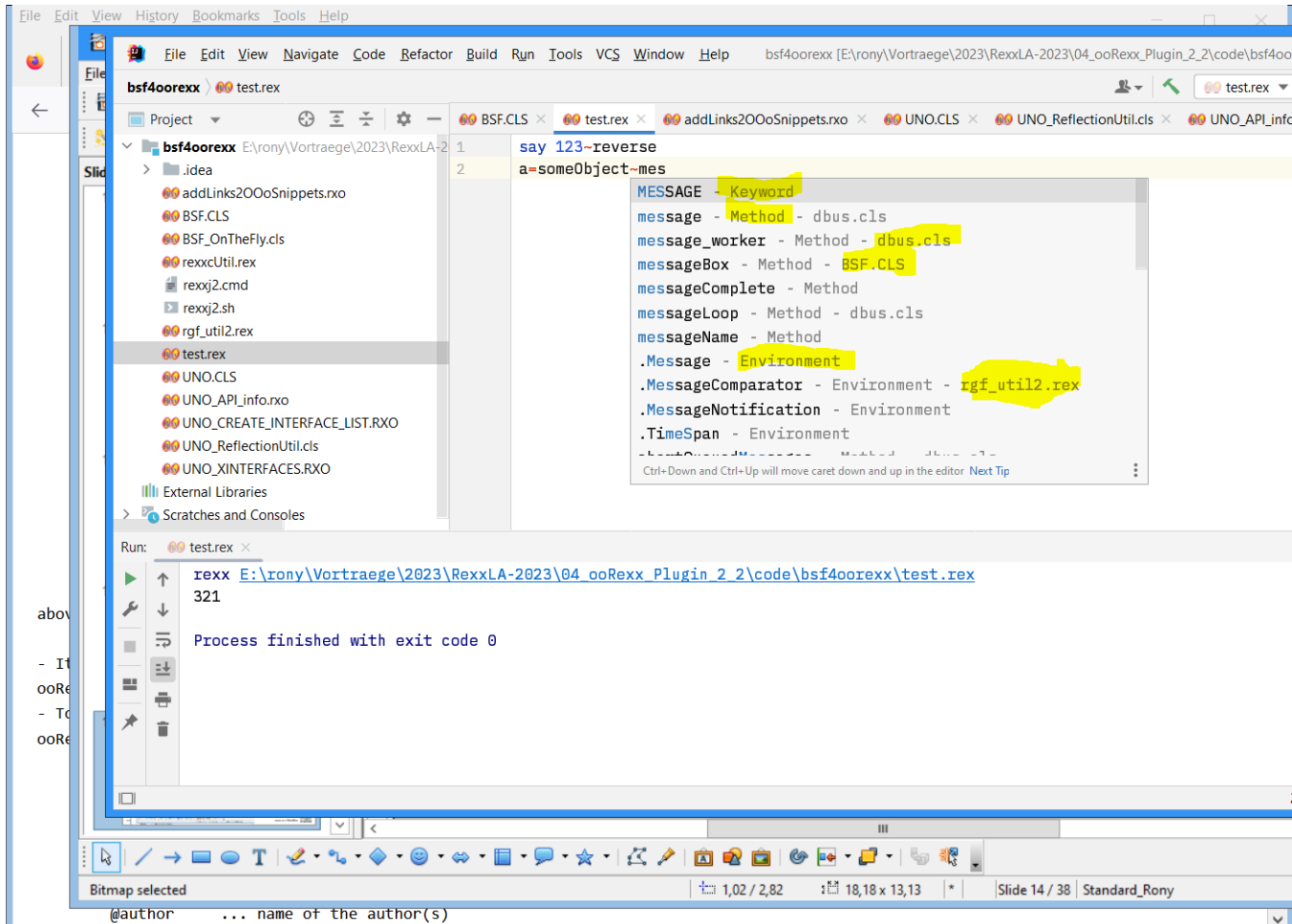
- run - Method
- runLater - Method - BSF.CLS
- runLaterLatest - Method - BSF.CLS
- runLaterLatestPush - Method - BSF.CLS
- runLaterPush - Method - BSF.CLS
- .true - Environment
- TRUNC - Function
- trunc - Method

Below the editor, the 'Run' console shows the execution of the script:

```
Run: test.rexx x  
↑ rexx E:\rony\Vortraege\2023\RexxA-2023\04 ooRexx_Plugin_2_2\code\bsf4ooress\test.rexx  
321  
Process finished with exit code 0
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help), a toolbar, and a sidebar with a project tree. The status bar at the bottom indicates 'Slide 14 / 38 | Standard\_Rony'.

# ooRexxPlugin 2.2, Auto Completion, 4



# ooRexxDoc 2.2

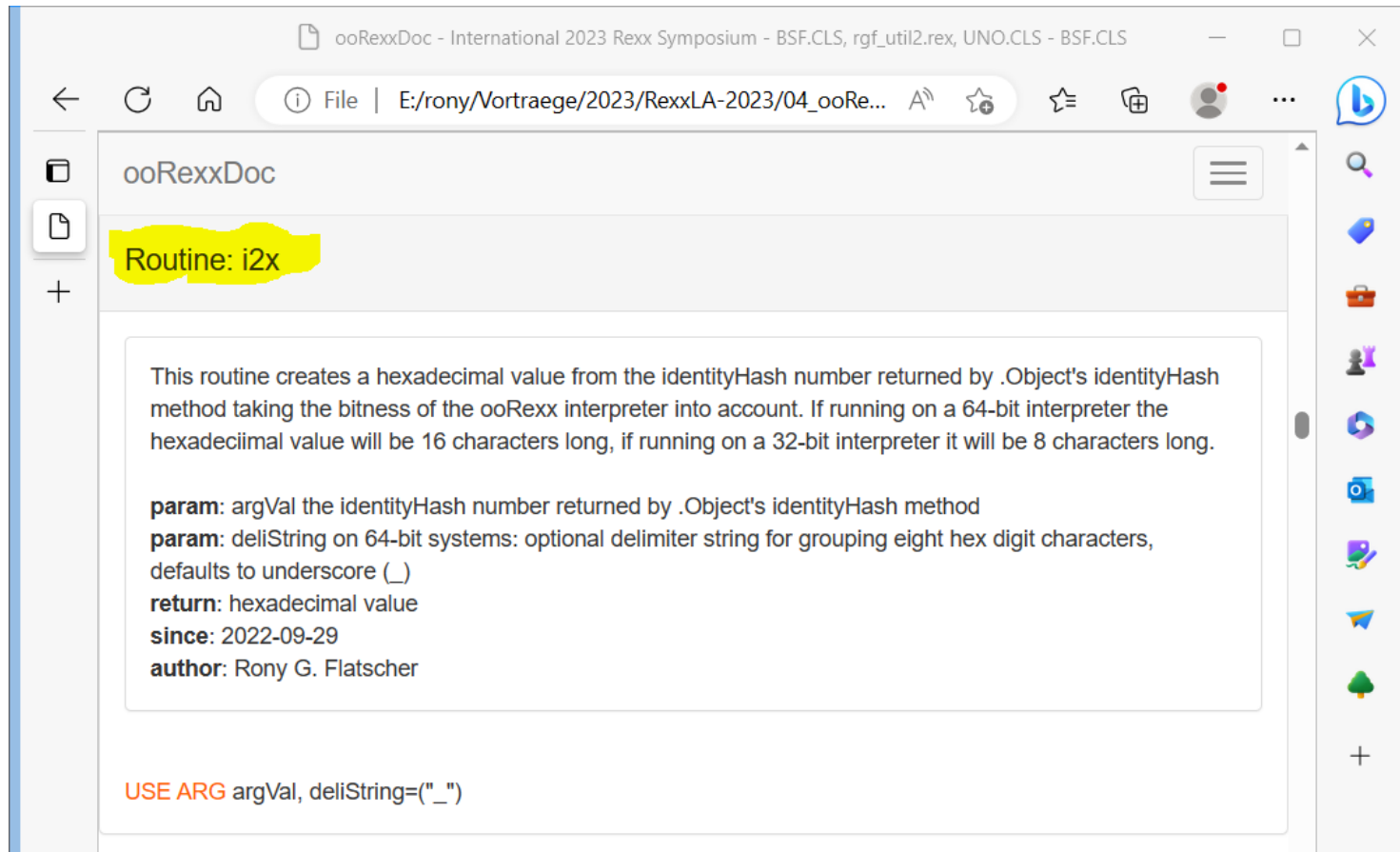
- Modelled after JavaDoc
  - Creates HTML documentation from any Rexx, ooRexx and executor programs
  - Rexx programs may include ooRexxDoc annotations in ooRexxDoc block comments
    - Led in with `/**` (note the second asterisk)
    - Processes ooRexxDoc annotations in ooRexxDoc comments
      - `@author`, `@condition`, `@deprecated`, `@param`, `@return`, `@see`, `@since`, `@version`
  - Can be run from the command line
- Download ooRexxDoc-2.2.0-GA.zip, unblock and unzip
  - Usage information in file [USAGE.md](#)
- Download URL
  - <https://sourceforge.net/projects/bsf4oorexx/files/Sandbox/aseik/ooRexxIDEA/GA/2.2.0/>

# IntelliJ ooRexxDoc, 1

```
6328
6329
6330 /** This routine creates a hexadecimal value from the identityHash number returned by .Object's
6331     identityHash method taking the bitness of the ooRexx interpreter into account.
6332     If running on a 64-bit interpreter the hexadecimal value will be 16 characters
6333     long, if running on a 32-bit interpreter it will be 8 characters long.
6334
6335     @param argVal the identityHash number returned by .Object's identityHash method
6336     @param deliString on 64-bit systems: optional delimiter string for grouping eight hex digit
6337         characters, defaults to underscore (_)
6338     @return hexadecimal value
6339     @since 2022-09-29
6340     @author Rony G. Flatscher
6341 */
6342 ::routine i2x public -- "pointer" to hexadecimal string: convert identityHash value to hexadecimal string
6343 use strict arg argVal, deliString=_(" ")
6344
6345 iDigits=.rexxinfo~internalDigits -- get number of internal digits
6346 numeric digits iDigits
6347 val=argVal~d2x(iDigits) -- convert to a hexadecimal string
6348 len=(iDigits=9)~?(8,16) -- determine length of hexadecimal value
6349 hexval = val~right(len) -- extract the hex digits representing 32 or 64 bit value
6350 if iDigits=18, deliString<>" then -- insert delimiter into hexadecimal string
6351     hexVal=hexVal~insert(deliString,8)
6352 return hexVal
6353
6354
6355 /** Routine that makes sure that a RexxAndJava (RexxEngine) instance gets initialized
6356 * properly, even if the prolog code does not get executed (to be expected in future
```



# IntelliJ ooRexxDoc, 2



The screenshot shows the IntelliJ IDEA interface with the ooRexxDoc tool window open. The title bar of the tool window reads "ooRexxDoc - International 2023 Rexx Symposium - BSF.CLS, rgf\_util2.rex, UNO.CLS - BSF.CLS". The main content area displays the documentation for the "Routine: i2x" routine, which is highlighted in yellow. The documentation includes a description of the routine's purpose, parameter details, return value, and author information. At the bottom, there is a note about using ARG.

ooRexxDoc

Routine: i2x

This routine creates a hexadecimal value from the identityHash number returned by .Object's identityHash method taking the bitness of the ooRexx interpreter into account. If running on a 64-bit interpreter the hexadecimal value will be 16 characters long, if running on a 32-bit interpreter it will be 8 characters long.

**param:** argVal the identityHash number returned by .Object's identityHash method  
**param:** deliString on 64-bit systems: optional delimiter string for grouping eight hex digit characters, defaults to underscore (\_)  
**return:** hexadecimal value  
**since:** 2022-09-29  
**author:** Rony G. Flatscher

USE ARG argVal, deliString=\_(" ")

# IntelliJ ooRexxDoc, 3

The screenshot shows the IntelliJ IDEA interface with the following elements:

- Project View:** Shows the project structure for 'bsf4oorexx' with files like 'BSF.CLS', 'test.rexx', and various utility files.
- Code Editor:** Displays ooRexx code for the 'i2x' routine. A comment block is highlighted in yellow, describing the routine's purpose: `/** This routine creates a hexadecimal value from the identityHash number returned by .Object's identityHash method taking the bitness of the ooRexx interpreter into account. If running on a 64-bit interpreter the hexadecimal value will be 16 characters long, if running on a 32-bit interpreter it will be 8 characters long. */`
- Context Menu:** A right-click context menu is open over the comment. The options 'Create 'ooRexxDoc' Documentation ...' and 'Quick Create 'ooRexxDoc' Documentation ...' are highlighted in yellow.
- Code Snippets:** Below the comment, the routine's parameters and return value are defined: `@param argVal the identityHash value returned by .Object's identityHash method for grouping eight hex digits or for grouping four hex digits on 64-bit systems. @param deliString on 64-bit systems, default is 'x' on 32-bit systems, default is 'X'. @return hexadecimal value @since 2022-09-29 @author Rony G. Flatscher`
- Code Implementation:** The routine body starts with `::routine i2x public -- "public" use strict arg argVal, deliString` and includes logic for determining the number of digits and inserting delimiters.

# IntelliJ ooRexxDoc, 4

ooRexxDoc Settings

Title: `International 2023 Rexx Symposium - BSF.CLS, rgf_util2.rex, UNO.CLS`

Input

File

`file://E:/rony/Vortraege/2023/RexxLA-2023/04_ooRexx_Plugin_2_2/code/bsf4ooorexx/BSF.CLS`  
`file://E:/rony/Vortraege/2023/RexxLA-2023/04_ooRexx_Plugin_2_2/code/bsf4ooorexx/rgf_util2.rex`  
`file://E:/rony/Vortraege/2023/RexxLA-2023/04_ooRexx_Plugin_2_2/code/bsf4ooorexx/UNO.CLS`

Output: `file://E:/rony/Vortraege/2023/RexxLA-2023/04_ooRexx_Plugin_2_2`

Header Properties

Key	Value
<code>\${.now?datetime}</code>	Run-Date

Overview

Include

Public  Package  Private  Variable References

Use this configuration for quick generation  Open documentation after creation

OK Cancel

# IntelliJ ooRexxDoc, 5a (Firefox)

File Edit View History Bookmarks Tools Help

ooRexxDoc - International 2023 Re... +

file:///E:/rony/Vortraege/2023/RexxLA- Search

ooRexxDoc Home BSF.CLS Search Run Date: May 11, 2023, 12:42:55 PM

Packages

- BSF.CLS
- rgf\_util2.rex
- UNO.CLS

BSF.CLS Overview

- Routines `box` `box.strictArg` `bsf.compile` `bsf.contextClassLoader` `bsf.createArray` `bsf.createArrayOf` `bsf.createJavaArray` `bsf.createJavaArrayOf` `bsf.createProxyClass` `bsf.getConstant` `bsf.getConstantJSO` `bsf.getSlotDir` `bsf.getStaticValue` `bsf.getStaticValueJSO` `bsf.getStaticValueStrict` `bsf.getStaticValueStrictJSO` `bsf.haltThread` `bsf.iconv` `bsf.import` `bsf.importClass` `bsf.initRAJinstance` `bsf.isA` `bsf.JSOasUtf8string` `bsf.loadClass` `bsf.lookupBean` `bsf.pollEventText` `bsf.postEventText` `bsf.redirectTo` `bsf.stringToUtf8` `BSF.terminateRexxEngine` `bsf.unregisterBean` `bsf.utf8stringAsJSO` `bsf.utf8ToString` `bsf.version` `bsf.wrap` `bsf.wrapStaticFields` `create_display.version_entry` `escapeQuotes` `getAndSetPathTolcons` `i2x` `iif` `initialize.BSF.dir` `makeBSF4ooRexxLookLikeMacOSX` `parseJavaOptions` `pp` `ppJavaExceptionChain` `unbox` `unbox_ori` `xBSF` `xBsfCreateRexxProxy` `xBsfLoadJava` `xBsfRexxProxy` `xBsfUninit4JavaBean`
- Floating Methods `newStrict`
- Classes
  - `AbstractGUIThread`
    - Attributes: `JavaGuiUtilityClz` `msgQueue` `runnable` `waitingToRun`
    - Methods: `init` `createGUIMessage` `invokeOnGuiThread` `isGuiThread` `process` `removeMessage` `run` `runLater` `runLaterLatest` `runLaterLatestPush` `runLaterPush` `setup`
  - `AwtGUIThread`
    - Attributes: `JavaGuiUtilityClz` `msgQueue` `runnable` `waitingToRun`
    - Methods: `init` `invokeOnGuiThread` `isGuiThread` `setup`
  - `BSF`

# IntelliJ ooRexxDoc, 5b (Edge)

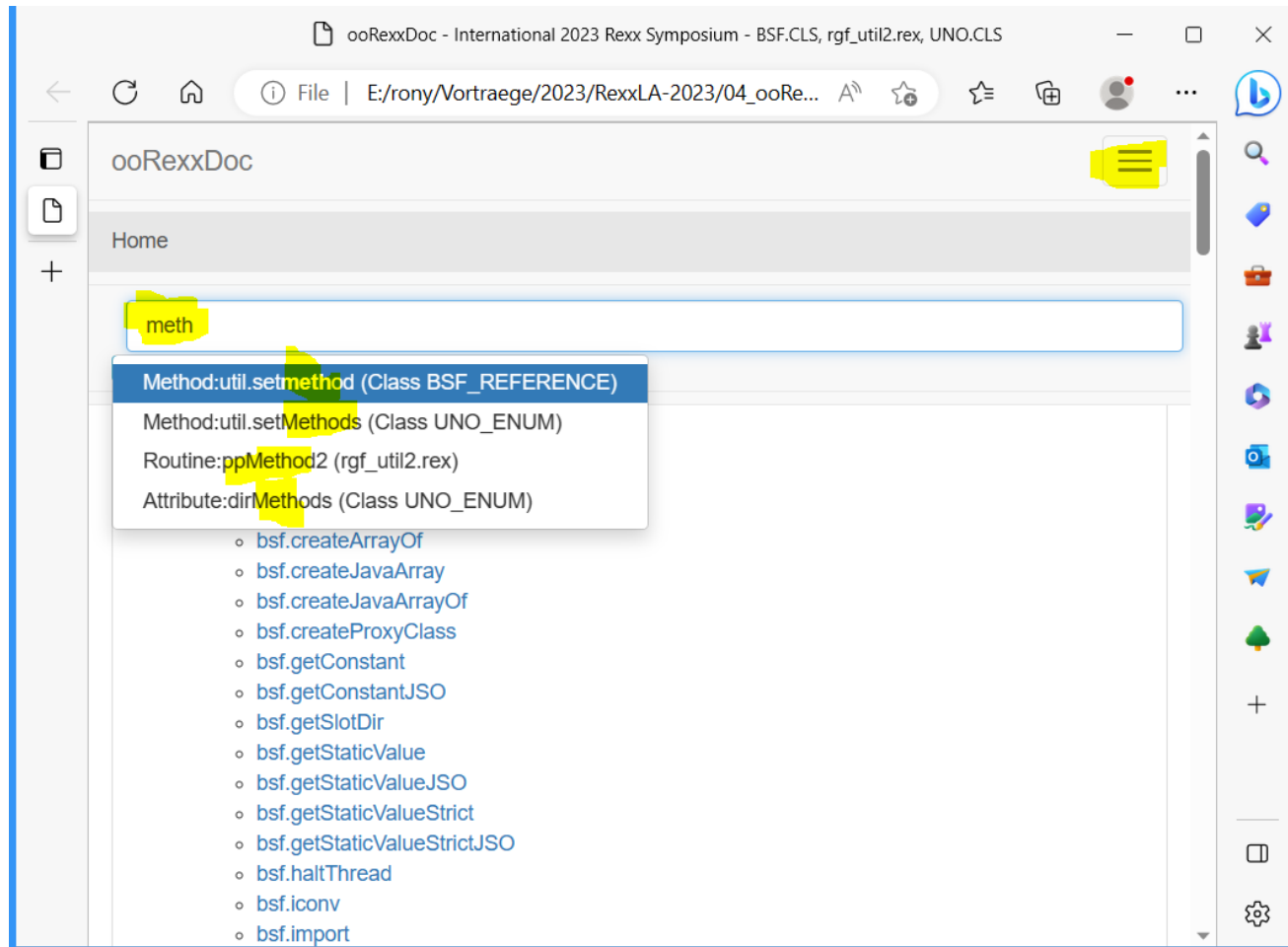
ooRexxDoc - International 2023 Rexx Symposium, ooRexxDoc BSF.CLS, UNO.CLS, rgf\_util2.rex

File | E:/rony/Vortraege/2023/RexxLA-2023/04\_ooRe...

ooRexxDoc

- BSF.CLS
  - Routines
    - box
    - box.strictArg
    - bsf.compile
    - bsf.contextClassLoader
    - bsf.createArray
    - bsf.createArrayOf
    - bsf.createJavaArray
    - bsf.createJavaArrayOf
    - bsf.createProxyClass
    - bsf.getConstant
    - bsf.getConstantJSO
    - bsf.getSlotDir
    - bsf.getStaticValue
    - bsf.getStaticValueJSO
    - bsf.getStaticValueStrict
    - bsf.getStaticValueStrictJSO
    - bsf.haltThread
    - bsf.iconv
    - bsf.import

# IntelliJ ooRexxDoc, 6



# IntelliJ ooRexxDoc, 7

The screenshot shows the IntelliJ ooRexxDoc interface. The browser address bar indicates the file path: `E:/rony/Vortraege/2023/RexxLA-2023/04_ooRe...`. The main content area displays the following information:

- Class:** BSF\_REFERENCE (highlighted in yellow)
- Sub-Class:** BSF
- Attributes:** No Attributes
- Methods:**
  - `init` (with `USE ARG` and parameters `beanName, blncr=false`)
  - `util.setmethod` (highlighted in yellow)
- Constants:** No Constants

A "Back to Top" button is visible at the bottom right of the content area.

# IntelliJ ooRexxDoc, 8

ooRexxDoc - International 2023 REXX Symposium, ooRexxDoc BSF.CLS, UNO.CLS, rgf\_util2.rex - BSF...

File | E:/rony/Vortraege/2023/RexxLA-2023/04... | Search

Run-Date: May 11, 2023, 12:00:18 PM

ooRexxDoc Home BSF.CLS

Packages

- BSF.CLS
- rgf\_util2.rex
- UNO.CLS

BSF.CLS Overview

- Routines `box` `box.strictArg` `bsf.compile` `bsf.contextClassLoader` `bsf.createArray` `bsf.createArrayOf` `bsf.createJavaArray` `bsf.createJavaArrayOf` `bsf.createProxyClass` `bsf.getConstant` `bsf.getConstantJSO` `bsf.getSlotDir` `bsf.getStaticValue` `bsf.getStaticValueJSO` `bsf.getStaticValueStrict` `bsf.getStaticValueStrictJSO` `bsf.haltThread` `bsf.iconv` `bsf.import` `bsf.importClass` `bsf.initRAInstance` `bsf.isA` `bsf.JSOasUtf8string` `bsf.loadClass` `bsf.lookupBean` `bsf.pollEventText` `bsf.postEventText` `bsf.redirectTo` `bsf.stringToUtf8` `BSF.terminateRexxEngine` `bsf.unregisterBean` `bsf.utf8stringAsJSO` `bsf.utf8ToString` `bsf.version` `bsf.wrap` `bsf.wrapStaticFields` `create_display.version_entry` `escapeQuotes` `getAndSetPathToIcons` `i2x` `iif` `initialize.BSF.dir` `makeBSF4ooRexxLookLikeMacOSX` `parseJavaOptions` `pp` `pp.JavaExceptionChain` `unbox` `unbox_ori` `xBSF` `xBsfCreateRexxProxy` `xBsfLoadJava` `xBsfRexxProxy` `xBsfUninit4.JsBean`
- Floating Methods `newStrict`



# IntelliJ ooRexxDoc, 9

ooRexxDoc - International 2023 Rexx Symposium - BSF.CLS, rgf\_util2.rex, UNO.CLS - BSF.CLS

File | E:/rony/Vortraege/2023/RexxLA-2023/04\_ooRe...

ooRexxDoc

Class: AwtGUIThread

Sub-Class: AbstractGUIThread

This class is meant to help ooRexx programmers to simplify interactions with the awt/swing GUI. If Rexx code is invoked by the GUI maintenance code, that Rexx code will already execute on the GUI thread (called "event dispatch thread" a.k.a. "awt thread") and can successfully interact with the GUI and the elements. However, if Rexx code on a non-GUI-thread wishes to interact with the GUI, this would hang the GUI (single threaded implementations are typical for GUIs). Rather, such an interaction must occur on the GUI thread.

The class methods `runLater` and `runLaterLatest` allow Rexx programmers to send Rexx messages later, when running in the GUI thread.

since: 2018-07-14

Attributes

- `JavaGuiUtilityClz` GETTER CLASS
- `msgQueue` GETTER CLASS
- `runnable` GETTER CLASS

Back to Top

# IntelliJ ooRexxDoc, 10

The screenshot shows the IntelliJ ooRexxDoc application interface. The browser address bar indicates the file path: `E:/rony/Vortraege/2023/RexxLA-2023/04_ooRe...`. The application title is "ooRexxDoc". The main content area displays the documentation for the class `AbstractGUIThread`.

**Class: AbstractGUIThread**

This abstract class implements the protocol and common methods for the awt/swing and JavaFX utility classes that allow one to interact with the GUI objects on the awt/JavaFX thread.

If Rexx code is invoked by the GUI maintenance code in form of callbacks, that Rexx code will already execute on the GUI thread and can successfully interact with the GUI and the elements. However, if Rexx code on a non-GUI-thread wishes to interact with the GUI, this would hang the GUI (single threaded implementations are typical for GUIs). Rather, such an interaction must occur on the GUI thread.

The class methods `runLater` and `runLaterLatest` allow Rexx programmers to send Rexx messages later, when running in the GUI thread.

**since:** 2017-08-14

**Attributes**

- `JavaGuiUtilityClz` **GETTER** **CLASS** **ABSTRACT**
- `msgQueue` **GETTER** **SETTER** **CLASS** **ABSTRACT**
- `runnable` **GETTER** **SETTER** **CLASS** **ABSTRACT**
- `waitingToRun` **GETTER** **SETTER** **CLASS** **ABSTRACT**

A "Back to Top" button is visible at the bottom right of the content area.

# Roundup

- ooRexxPlugin 2.2
  - Adds ooRexx, Rexx and executor support to JetBrains's IntelliJ IDEA
  - Makes IntelliJ IDEA features available for ooRexx, Rexx and executor programmers
  - Introduces ooRexx, Rexx and executor auto name completion
- ooRexxDoc
  - Creates ooRexxDoc documentation from Rexx, ooRexx and executor programs
    - Right mouse click in IntelliJ
    - Stand-alone version can also be invoked via the command line for software projects that need to create ooRexxDoc documentation

# Links (As of 2023-05-11)

- [JetBrains' IntelliJ IDEA](https://www.jetbrains.com/idea/download): <https://www.jetbrains.com/idea/download>
- [ooRexxPlugin](https://sourceforge.net/projects/bsf4oorexx/files/Sandbox/aseik/ooRexxIDEA/GA/2.2.0/): <https://sourceforge.net/projects/bsf4oorexx/files/Sandbox/aseik/ooRexxIDEA/GA/2.2.0/>
- [ooRexxDoc](https://sourceforge.net/projects/bsf4oorexx/files/Sandbox/aseik/ooRexxIDEA/GA/2.2.0/): <https://sourceforge.net/projects/bsf4oorexx/files/Sandbox/aseik/ooRexxIDEA/GA/2.2.0/>
  - Completion add-ons
    - <https://sourceforge.net/projects/bsf4oorexx/files/Sandbox/aseik/ooRexxIDEA/GA/2.2.0/>
  - Since 2023-05-11: [extractNamesFromRexxPackages.rex](#) and [rgfNames.zip](#)
- [ooRexx 5.1](https://sourceforge.net/projects/oorexx/files/oorexx/5.1.0beta/): <https://sourceforge.net/projects/oorexx/files/oorexx/5.1.0beta/>
- [Java/OpenJDK with JavaFX](https://www Azul.com/downloads/?package=jdk#zulu) modules, e.g. <https://www.azul.com/downloads/?package=jdk#zulu>
- [BSF4ooRexx850](https://sourceforge.net/projects/bsf4oorexx/files/beta/20221004/): <https://sourceforge.net/projects/bsf4oorexx/files/beta/20221004/>
- [executor](https://github.com/jlfaucher/executor): <https://github.com/jlfaucher/executor>