

# Package ‘JGR’

May 20, 2023

**Version** 1.9-2

**Title** Java GUI for R

**Date** 2023-05-20

**Author** Markus Helbig <mhelbig81@gmail.com>, Simon Urbanek, Ian Fellows

**Maintainer** Markus Helbig <mhelbig81@gmail.com>

**SystemRequirements** Java (>= 8)

**Depends** R (>= 3.0.0), rJava (>= 1.0), JavaGD (>= 0.6)

**Suggests** XLConnect

**Description** Java GUI for R - cross-platform, universal and unified Graphical User Interface for R. For full functionality on Windows and Mac OS X JGR requires a start application which depends on your OS.

**License** GPL-2

**URL** <https://rforge.net/JGR/>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-05-20 09:50:02 UTC

## R topics documented:

jgr.addMenu . . . . .	2
jgr.addMenuItem . . . . .	3
jgr.addMenuSeparator . . . . .	4
jgr.addSubMenu . . . . .	4
jgr.getMenuNames . . . . .	5
jgr.internals . . . . .	6
jgr.removeMenu . . . . .	6
jgr.supplements . . . . .	7
launchJGR . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

jgr.addMenu	<i>New JGR Console Menu</i>
-------------	-----------------------------

---

### Description

adds a new Menu to MenuBar in JGR Console

### Usage

```
jgr.addMenu(name)
jgr.insertMenu(name, index)
```

### Arguments

name	Menu name
index	index at which to insert

### Value

Menu

### See Also

[jgr.addMenuItem](#) [jgr.addMenuSeparator](#)

### Examples

```
jgr.addMenu("Workspace")
jgr.addMenuItem("Workspace", "Browse", "ls()")
jgr.addMenuSeparator("Workspace")
jgr.addMenuItem("Workspace", "Browse (pos=2)", "ls(pos=2)")

menus <- jgr.getMenuNames()
index <- which(menus=="Packages & Data")
if(length(index)==0) index <- 1

jgr.insertMenu("User menu", index)
jgr.addMenuItem("User menu", "Good place for user generated menus is",
               "print('before Packages & Data')")
```

---

jgr.addMenuItem	<i>New JGR Console MenuItem</i>
-----------------	---------------------------------

---

**Description**

adds a new MenuItem to specified JGR Console Menu

**Usage**

```
jgr.addMenuItem(menu, name, command, silent=TRUE)
jgr.insertMenuItem(menu, name, command, index, silent=TRUE)
```

**Arguments**

menu	Name of the menu that this item will be added to
name	Name of the menu item to add
command	R expression(s) as a string to be parsed and evaluated or a function that will be called (without arguments) when the menu item is selected
index	index at which to insert
silent	If FALSE, executes as if entered into the cons

**Value**

MenuItem

**See Also**

[jgr.addMenu](#) [jgr.addMenuSeparator](#)

**Examples**

```
jgr.addMenu("Workspace")
jgr.addMenuItem("Workspace", "Browse", "ls()", FALSE)
jgr.addMenuSeparator("Workspace")
jgr.addMenuItem("Workspace", "List Functions",
  function() unlist(lapply(ls(envir=.GlobalEnv),
    function(x) if (is.function(get(x))) x else NULL )))
```

---

`jgr.addMenuSeparator`    *New JGR Console MenuSeparator*

---

**Description**

adds a Separator to a Menu in JGR Console

**Usage**

```
jgr.addMenuSeparator(menu)
jgr.insertMenuSeparator(menu, index)
```

**Arguments**

menu	name of the menu to add this separator to
index	index at which to insert

**Value**

Menu

**See Also**

[jgr.addMenu](#) [jgr.addItem](#)

**Examples**

```
jgr.addMenu("Workspace")
jgr.addItem("Workspace", "Browse", "ls()")
jgr.addMenuSeparator("Workspace")
jgr.addItem("Workspace", "Browse (pos=2)", "ls(pos=2)")
```

---

`jgr.addSubMenu`    *New JGR Console sub-menu*

---

**Description**

adds a sub-menu to a Menu in JGR Console

**Usage**

```
jgr.addSubMenu(menu, subMenuName, labels, commands)
jgr.insertSubMenu(menu, subMenuName, labels, commands, index)
```

**Arguments**

menu	name of the menu to add this sub-menu to
subMenuName	name of the sub-menu
labels	a character vector of menu items
commands	a character vector of commands to be executed
index	index at which to insert

**Value**

Menu

**See Also**

[jgr.addMenu](#) [jgr.addItem](#)

**Examples**

```
jgr.addMenu("added menu")
jgr.addItem("added menu", "print 1","print(1)")
jgr.insertMenuItem("added menu","print 1 as if entered in console","print(1)",1,FALSE)
jgr.addMenuSeparator("added menu")
jgr.addSubMenu("added menu","sub menu",c("a","b","c"),c("print('a')","print('b')","print('c')"))
```

---

`jgr.getMenuNames`

*JGR Console menus*

---

**Description**

returns the menu / menu item names

**Usage**

```
jgr.getMenuNames()
jgr.getMenuItemNames(menu)
```

**Arguments**

menu	name of the menu
------	------------------

**Value**

a character vector

**See Also**

[jgr.addMenu](#) [jgr.addItem](#)

**Examples**

```

menus <- jgr.getMenuNames()
index <- which(menus=="Packages & Data")
if(length(index)==0) index <- 1

jgr.insertMenu("User menu",index)
jgr.addMenuItem("User menu", "Good place for user generated menus is",
               "print('before Packages & Data')")

```

---

<code>jgr.internals</code>	<i>Internal JGR functions</i>
----------------------------	-------------------------------

---

**Description**

The following functions are publicly visible, but with the exception of JGR they should be only invoked from the GUI and not directly by the user.

---

<code>jgr.removeMenu</code>	<i>remove JGR Console menus</i>
-----------------------------	---------------------------------

---

**Description**

removes a menu / menu item

**Usage**

```

jgr.removeMenu(index)
jgr.removeMenuItem(menu, index)

```

**Arguments**

<code>menu</code>	name of the menu
<code>index</code>	index of the item to remove

**See Also**

[jgr.addMenu](#) [jgr.addMenuItem](#)

**Examples**

```

jgr.addMenu("added menu")
jgr.addMenuItem("added menu", "print 1", "print(1)")
jgr.insertMenuItem("added menu", "print 1 as if entered in console", "print(1)", 1, FALSE)
jgr.addMenuSeparator("added menu")
jgr.addSubMenu("added menu", "sub menu", c("a", "b", "c"), c("print('a')", "print('b')", "print('c')"))

jgr.removeMenuItem("added menu", 1)
jgr.removeMenu(length(jgr.getMenuNames()))

```

## Description

The following functions are publicly visible, but with the exception of JGR they are usually invoked from the GUI and rarely directly by the user.

JGR can be used to update JGR including dependent packages or to start JGR from within R. JGR is always started as an independent application and the preferred way to start JGR is using launchers that are provided from the JGR website <http://www.rosuda.org>

## Usage

```
JGR(update = FALSE, javaArgs = NULL, jgrArgs = "")
broken.gomp()
installPackages(contriburl = NULL, type = getOption("pkgType"))
jgr.pager(file, header, title, delete.file)
jgr.browser(url, ...)
jgr.set.options(..., useJavaGD = TRUE, useJGRpager = TRUE,
                useJGRbrowser = TRUE, useHTMLHelp = TRUE)
object.browser()
package.manager()
reformat.code(txt)
```

## Arguments

update	If set to TRUE then JGR is updated for the RoSuDa repository (internet connection is required).
javaArgs	Arguments for Java.
jgrArgs	Arguments passed to the main method of JGR.
contriburl	same as in <a href="#">install.packages</a>
type	JGR uses the package type set in the R options
file	file to show
header	header for each file
title	overall title for the display
delete.file	should the files be deleted after display? Used for temporary files.
url	url to show
useJavaGD	if set to TRUE, JavaGD is set as the default device
useJGRpager	if set to TRUE, jgr.pager function is used as the default pager
useJGRbrowser	if set to TRUE, jgr.browser function is used as the default browser
useHTMLHelp	if set to TRUE, HTML help used as the default
...	additional parameters or options to set
txt	R code to be made pretty

---

launchJGR	<i>Creates a new process launching the JGR Console</i>
-----------	--

---

**Description**

Creates a new process launching the JGR Console

**Usage**

```
launchJGR(javaArgs=NULL, jgrArgs="", popMsgs=TRUE)
```

**Arguments**

javaArgs	Arguments for Java.
jgrArgs	Arguments passed to the main method of JGR.
popMsgs	If true, system pop-up messages are displayed if any errors are encountered.

**Value**

The system call for launching JGR



# Index

## \* programming

- jgr.addMenu, 2
- jgr.addMenuItem, 3
- jgr.addMenuSeparator, 4
- jgr.addSubMenu, 4
- jgr.getMenuNames, 5
- jgr.internals, 6
- jgr.removeMenu, 6
- jgr.supplements, 7
- .completeCommand (jgr.internals), 6
- .generate.mac.script (jgr.internals), 6
- .generate.run.script (jgr.internals), 6
- .generate.windows.script (jgr.internals), 6
- .getContent (jgr.internals), 6
- .getDataObjects (jgr.internals), 6
- .getFunctionsInWS (jgr.internals), 6
- .getModels (jgr.internals), 6
- .getOtherObjects (jgr.internals), 6
- .jgr.pkg.path (jgr.internals), 6
- .jgr.register.function (jgr.internals), 6
- .jgr.save.JavaGD.as (jgr.internals), 6
- .jgr.works (jgr.internals), 6
- .refresh (jgr.internals), 6
- .refreshKeyWords (jgr.internals), 6
- .refreshObjects (jgr.internals), 6

broken.gomp (jgr.supplements), 7

install.packages, 7

installPackages (jgr.supplements), 7

JGR (jgr.supplements), 7

jgr.addMenu, 2, 3–6

jgr.addMenuItem, 2, 3, 4–6

jgr.addMenuSeparator, 2, 3, 4

jgr.addSubMenu, 4

jgr.browser (jgr.supplements), 7

jgr.getMenuItemNames

(jgr.getMenuNames), 5

jgr.getMenuNames, 5

jgr.insertMenu (jgr.addMenu), 2

jgr.insertMenuItem (jgr.addMenuItem), 3

jgr.insertMenuSeparator

(jgr.addMenuSeparator), 4

jgr.insertSubMenu (jgr.addSubMenu), 4

jgr.internals, 6

jgr.pager (jgr.supplements), 7

jgr.removeMenu, 6

jgr.removeMenuItem (jgr.removeMenu), 6

jgr.set.options (jgr.supplements), 7

jgr.supplements, 7

launchJGR, 8

object.browser (jgr.supplements), 7

package.manager (jgr.supplements), 7

reformat.code (jgr.supplements), 7