

# Package ‘svGUI’

April 16, 2021

**Type** Package

**Version** 1.0.1

**Date** 2021-04-16

**Title** SciViews - Manage GUIs in R

**Description** The 'SciViews' 'svGUI' package eases the management of Graphical User Interfaces (GUI) in R. It is independent from any particular GUI widgets ('Tk', 'Gtk2', native, ...). It centralizes info about GUI elements currently used, and it dispatches GUI calls to the particular toolkits in use in function of the context (is R run at the terminal, within a 'Tk' application, a HTML page?).

**Maintainer** Philippe Grosjean <phgrosjean@sciviews.org>

**Depends** R (>= 2.6.0)

**Suggests** covr, knitr, markdown, testthat, spelling

**License** GPL-2

**URL** <https://github.com/SciViews/svGUI>, <https://www.sciviews.org/svGUI/>

**BugReports** <https://github.com/SciViews/svGUI/issues>

**RoxygenNote** 7.1.0

**VignetteBuilder** knitr

**Encoding** UTF-8

**Language** en-US

**NeedsCompilation** no

**Author** Philippe Grosjean [aut, cre] (<<https://orcid.org/0000-0002-2694-9471>>)

**Repository** CRAN

**Date/Publication** 2021-04-16 18:10:02 UTC

## R topics documented:

dont_ask . . . . .	2
gui . . . . .	3
gui_add . . . . .	4
setUI . . . . .	6

---

dont_ask	<i>Can we interrupt R to ask something to the user though the GUI?</i>
----------	--

---

### Description

Determine if R code execution can be interrupted by the GUI, e.g., using a modal dialog box. It depends both on R being in `interactive()` mode and the ask flag of the GUI being set to TRUE.

### Usage

```
dont_ask(gui = .GUI)
```

```
dontAsk(gui = .GUI)
```

### Arguments

`gui`            A gui object.

### Details

`dontAsk` and `dont_ask` are aliases.

### Value

TRUE if the GUI cannot interrupt R. The function triggering the dialog box should then not be displayed and it should return the default value as the result. The function returns TRUE if R is run in a non interactive session, or if ask is set to FALSE for the GUI, or if it is not specified (ask is NULL) then `getOptions("gui.ask")` is used.

### See Also

[gui\\_ask\(\)](#), [gui](#)

### Examples

```
# What is the current state for the default GUI?  
dont_ask()
```

---

gui	<i>A GUI object.</i>
-----	----------------------

---

### Description

The gui object contains and manages GUI-related data.

### Usage

```
## S3 method for class 'gui'  
gui$x  
  
## S3 method for class 'gui'  
print(x, ...)  
  
is.gui(x)
```

### Arguments

gui	A gui object..
x	An object or a function for \$.
...	Further arguments (not used yet).

### See Also

[gui\\_add\(\)](#)

### Examples

```
# Create a GUI  
gui_add("myGUI")  
is.gui(myGUI)  
myGUI  
# Put an object in the GUI environment (fake button)  
myGUI$button <- "my_button"  
# Retrieve it  
myGUI$button  
# Get the current status of the GUI  
myGUI$status  
# Eliminate this GUI and all its objects  
gui_remove("myGUI")
```

---

`gui_add`*Creation and management of GUI objects.*

---

**Description**

Create and manipulate gui objects to manage 'SciViews'-compatible GUIs (Graphical User Interfaces).

**Usage**

```
gui_add(gui.name = ".GUI", widgets = c("nativeGUI", "textCLI"), ask)
```

```
guiAdd(gui.name = ".GUI", widgets = c("nativeGUI", "textCLI"), ask)
```

```
gui_change(  
  gui.name = ".GUI",  
  widgets = c("nativeGUI", "textCLI"),  
  reset = FALSE,  
  ask  
)
```

```
guiChange(  
  gui.name = ".GUI",  
  widgets = c("nativeGUI", "textCLI"),  
  reset = FALSE,  
  ask  
)
```

```
gui_remove(gui.name)
```

```
guiRemove(gui.name)
```

```
gui_list()
```

```
guiList()
```

```
gui_widgets(gui, gui.name = ".GUI")
```

```
guiWidgets(gui, gui.name = ".GUI")
```

```
gui_widgets(x, reset = FALSE) <- value
```

```
guiWidgets(x, reset = FALSE) <- value
```

```
gui_ask(gui.or.name, ask)
```

```
guiAsk(gui.or.name, ask)
```

```
gui_ask(x) <- value
```

```
guiAsk(x) <- value
```

### Arguments

gui.name	The name of the GUI. It is also the name of the object stored in <code>SciViews:TempEnv</code> where you can access it.
widgets	The list of widgets that GUI uses, listed in a priority order.
ask	Logical indicating if modal dialog boxes should be display ( <code>ask = TRUE</code> ), or if those dialog boxes are by-passed, using default values to simulate script running in non interactive mode, or to test scripts without interruption, using only provided default values (useful for automated tests).
reset	Should the GUI's main parameters ( <code>widgets</code> , <code>ask</code> ) be reset to default values?
gui	A gui object. If provided, it supersedes any value provided in <code>gui.name</code> .
x	A gui object.
value	The list of widgets to add to this GUI, in priority order, or should we change <code>ask</code> to <code>TRUE</code> , <code>FALSE</code> or <code>NULL</code> (then, use the default value stored in <code>getOption("gui.ask")</code> ).
gui.or.name	A gui object or its name.

### See Also

[gui](#), [setUI\(\)](#), [dont\\_ask\(\)](#)

### Examples

```
# A 'gui' object named .GUI is automatically created in 'SciViews:TempEnv'
gui_list()

# Create a new GUI object to manage a separate GUI in the same R session
gui_add("myGUI")
gui_list()

# Change general properties of this GUI
gui_ask(myGUI) <- FALSE
# Add widgets to this GUI (you must provide methods for them)
# see the svDialogs package for examples
gui_widgets(myGUI) <- "tcltkWidgets"
gui_widgets(myGUI) # Added to existing ones if reset is FALSE

# Remove this new GUI
gui_remove("myGUI")
```



msg	The message that explains the status. Cannot be provided without status.
default	The default value to return if the UI is by-passed because in non interactive mode, or ask is FALSE.
msg.no.ask	The message that explains the status in case the UI is by-passed.

### Methods (by class)

- gui: Set an UI property for a gui object.
- gui: Start an UI for a gui object.

### See Also

[gui\\_add\(\)](#), [\\$.gui\(\)](#)

### Examples

```
# Imagine you implement a new input box
# In your function, you have this code:
myInput <- function(default = "an answer", gui = .GUI) {

  # Start a GUI action... or by-pass it!
  if (gui$startUI("myInput", call = match.call(), default = default,
    msg = "Displaying an input dialog box",
    msg.no.ask = "An input dialog box was by-passed")) {

    # Here the input dialog box is displayed and R waits for user feedback
    # ... [your code here]
    res <- "some results" # Imagine this is the text typed in the box

    # When the input dialog box is closed, the function should do:
    setUI(res = res, status = NULL)
  }
  invisible(gui)
}
```

# Index

## \* GUI API implementation

- dont\_ask, 2
- gui, 3
- gui\_add, 4
- setUI, 6

## \* misc

- dont\_ask, 2
- gui, 3
- gui\_add, 4
- setUI, 6

\$.gui(gui), 3

\$.gui(), 7

dont\_ask, 2

dont\_ask(), 5

dontAsk(dont\_ask), 2

gui, 2, 3, 5

gui\_add, 4

gui\_add(), 3, 7

gui\_ask(gui\_add), 4

gui\_ask(), 2

gui\_ask<- (gui\_add), 4

gui\_change(gui\_add), 4

gui\_list(gui\_add), 4

gui\_remove(gui\_add), 4

gui\_widgets(gui\_add), 4

gui\_widgets<- (gui\_add), 4

guiAdd(gui\_add), 4

guiAsk(gui\_add), 4

guiAsk<- (gui\_add), 4

guiChange(gui\_add), 4

guiList(gui\_add), 4

guiRemove(gui\_add), 4

guiWidgets(gui\_add), 4

guiWidgets<- (gui\_add), 4

is.gui(gui), 3

print.gui(gui), 3

setUI, 6

setUI(), 5

startUI(setUI), 6