Package ‘miniGUI’

February 20, 2015

Title  tkcl quick and simple function GUI.
Version 0.8.0
Date 2010-07-01
Depends R (>= 2.5.0), tcltk
Author Jorge Luis Ojeda Cabrera
Description quick and simple tkcl miniGUI to call functions.
Maintainer Jorge Luis Ojeda Cabrera <jojeda@unizar.es>
License GPL
Repository CRAN
Date/Publication 2012-09-04 07:08:11
NeedsCompilation no

R topics documented:

makeWidgetCmd .................................................. 1
mapFuncToWidget .................................................. 3
miniGUI ............................................................. 5
miniGUIhelpers .................................................... 7
miniGUIinputWidget ............................................... 8

Index ............................................................ 11

makeWidgetCmd  R functions to build a GUI window

Description
Function that wraps the result of mapFuncToWidget as an R function that pop us a widget representing the function.
Usage

```
makeWidgetCmd(frmTitle, fun, baseFrame=.TkRoot, STORE="ff", GRAB=TRUE)
```

Arguments

- `frmTitle` title of the GUI window.
- `fun` function to map.
- `baseFrame` `tcltk` parent frame of the GUI window for the function `fun`.
- `STORE` A string. Name of the place where to store details needed by the GUI to perform the execution.
- `GRAB` Logical. When `true` disable input in any other window.

Details

The main use of this function is to obtain a function that called creates a widget that allows the parameter input and execution of function `fun`. It also adds a `Quit` `fun` function to close the widget.

Value

This function returns an `R\ function.`

Author(s)

Jorge Luis Ojeda Cabrera (`<jojeda@unizar.es>`).

See Also

`miniGUI, makeWidgetCmd, miniGUIGetFormals, addMenusCmd, tcltk`.

Examples

```
require(tcltk)
##
## a simple example
##
g <- function(a=1, b=rnorm) {cat("--g--"); paste("g(a,b)="", a+b(a))}
h <- function(a=1, b=3, c=3) {cat("--h--"); paste("h(a,b,c)="", a+b+c)}
## create functions
gg <- makeWidgetCmd("Hay it is g !!", g, GRAB=FALSE)
hh <- makeWidgetCmd("Hay h here !!", h, GRAB=FALSE)
## calling them
gg()
cat("\n\nClose it before calling hh(), they sharer parameters a and b!!")
hh()
##
## simple example(continuation)
##
## to be able to use both at the same time: save info for h in other place
hh <- makeWidgetCmd("Hay h here !!", h, STORE="h")
```

```
gg()
```
mapFuncToWidget

Description

Function maps a large class of R functions onto a set of tcltk widgets that allows the input of its parameter.

Usage

mapFuncToWidget(f, frm, bttLabel="OK", STORE="ff", callSubst="mini GUI call")

Arguments

f  Function to map.
frm  tcltk frame to place the GUI window.
bttLabel  execution button label.
STORE  A string. Name of the place where to store details needed by the GUI to perform the execution.
callSubst  string to set call attribute/slot in some of the R computations results.

Details

This function returns a frame which contains pairs of tcltk labels and text entry (or any other tktcl widget that allows to input values) and a button. In this way, this function maps an R function f into a GUI window that allows its computation. Therefore, it provides a map from the a set of R function onto some class of GUI windows.

Usually, the way function is executed provides with nasty and long call attributes, call parameter substitute these allowing a much more comfortable output.

The string provided by STORE is used to store the function arguments in the list miniGUIData, enabling in this way the computation of the function.

Value

This function returns a tcltk frame (an object created with tkframe).

Author(s)

Jorge Luis Ojeda Cabrera (<jojeda@unizar.es>).

See Also

miniGUI, makeWidgetCmd, miniGUIGetFormals, addMenusCmd, tcltk.
Examples

```r
require(tcltk)
##
## a window for lm
##
## create some data (in the global environment)
n <- 100
d <- data.frame(x=runif(n))
d$z <- 0.5 * rnorm(n)
d$y <- 2 * d$x + d$z
## create a tcltk frame and give it a title
frm <- tktoplevel()
tkwm.title(frm,"mapFuncToWidget for lm")
## create the GUI window map of lm
mapFuncToWidget(lm,frm)
## ...you may close the window

##
## a window for T tests
##
myTtest <- function(x,y,mu=0) return( t.test(x=x,y=y,mu=mu) )
## create a tcltk frame and give it a title
frm <- tktoplevel()
tkwm.title(frm,"mapFuncToWidget for T tests")
## create the GUI window map of lm
mapFuncToWidget(myTtest,frm)
## ...you may close the window

##
## a simple example
##
g <- function(a=1,b=rnorm) {cat("--g--");paste("g(a,b)="",a+b(a))}
h <- function(a=1,b=3,c=3) {cat("--h--");paste("h(a,b,c)="",a+b+c)}
## create a tcltk frame and give it a title
frm <- tktoplevel()
tkwm.title(frm,"mapFuncToWidget for g")
## create the GUI window map of g
mapFuncToWidget(g,frm)
## ...you may close the window

##
## the use of STORE
##
frm <- tktoplevel()
tkwm.title(frm,"another map")
mapFuncToWidget(g,frm,bttLabel="press me !!!")
## ...and if you do not close the window
mapFuncToWidget(h,frm,bttLabel="exec h !!!",STORE="fff")
## ...STORE should be added because g and h shares parameter
```
## miniGUI

### Description

Function to create a simple Graphical User Interface based on R functions based on tcltk package.

### Usage

```r
miniGUI(mainFrameFun,opFuns=NULL,title="mini GUI",init=function() {},
        WRAPFUN=TRUE)
```
Arguments

- **mainFrameFun**: A function to display (params are labels and entry fields) in the main GUI window.
- **opfuns**: Named list of functions to add in the GUI menu Ops.
- **title**: Main window GUI title.
- **init**: Function to call before the GUI setup.
- **WRAPFUN**: when TRUE, the default option, an automatic tcltk widget is built for the functions in opfuns.

Details

*miniGUI* pops up a window widget with a menu bar containing two menus named **Basics** and **Ops** from which different functionality may be addressed during a *miniGUI* session. The menu **Basics** is used to request general purpose task during the session (like quitting), while **Ops** is usually where more specific tasks, those the GUI is devoted to and that are given in opfuns are grouped. When a menu item from **Ops** is selected a new window widget pops us reflecting all the parameters the function selected has, so that the user can fill text entries or set up the value for such parameters.

*{init}* can be used to add initialization and checking commands to the GUI. This function is executed before any other command.

When **WRAPFUN** is FALSE no tcltk widget is created for the functions in opfuns, allowing them to build their own widget. Do not use it unless functions encode its own tcltk, having into account the internals of the package to setup in a proper way the GUI for that specific function.

Value

*miniGUI* function returns nothing. Nevertheless, the results of the execution of the different functions called during the *miniGUI* session are available by means of the *miniGUI*ans object, and also by means of the **GUI ans.** entry in the menu **Basics**.

Author(s)

Jorge Luis Ojeda Cabrera (<jojeda@unizar.es>).

See Also

*miniGUI*, *makeWidgetCmd*, *addMenusCmd*, *tcltk*.

Examples

```r
require(tcltk)
#
## a simple example
##
fs <- list(  
  f=function(a=1) {cat("--f--");paste("f(a)="",a)},
  g=function(a=1,b=runif) {cat("--g--");paste("g(a,b)="",a+b(a))},
  h=function(a=1,b=3,c=3) {cat("--h--");paste("h(a,b,c)="",a+b+c)}  
)
```
## miniGUIhelpers

### Utility functions

**Description**

Some utility functions

**Usage**

- `addMenusCmd(cmdFuns, baseFrame)`
- `miniGUIgetFormals(f)`
- `miniGUIcallEval(f, p, e)`
- `evalPlugin(ev)`
- `miniGUIoutput(x, mess="\\miniGUI output: \\
")`

**Arguments**

- `cmdFuns` A list of R\(\) function.
- `baseFrame` `tcltk` parent frame of the GUI window for the function `f`.
- `f` An R\(\) function.
- `p` a list with all the parameters `f` requires.
miniGUIinputWidget

envi

environment where the parameters p of f are evaluated. By default, .GlobalEnv.

x
An R object to print.

mess
a string with a brief message that is printed before x.

ev
a string containing an R expression.

Details

These functions are internal functions that helps building the GUI map. addMenusCmd adds a menu to the main miniGUI frame. miniGUIgetFormals gets the parameter list of the function f filtering ellipsis. miniGUIcallEval performs the evaluation f when the arguments are set to those of p. miniGUIoutput is used to print out the result of the computation.

At the present moment, miniGUIeval is the same as the function miniGIUcallEval, while miniGUIans is used to store the result of the last computation made by a call to any of the miniGUI menu functions or any function widget created with by the functions makeWidgetCmd or mapFuncToWidget. miniGUIData, miniGUIffff and miniGUIans are used to store information or answers to commands that are required to compute function widgets. The functions setMiniGUIData setMiniGUIans, getMiniGUIData, getMiniGUIans are used to set and get data from miniGUIData and miniGUIans which lives in the miniGUIenvir environment.

Author(s)

Jorge Luis Ojeda Cabrera (<jojeda@unizar.es>).

See Also

miniGUI, makeWidgetCmd, addMenusCmd, tcltk.

---

miniGUIinputWidget

**Entry widgets**

Description

Function that builds different input methods.

Usage

miniGUIentry(x)
miniGUImenu(x)

Arguments

x
A R symbol, or numerical or character value. It can also be any R expression.

from, to, by
three numerical values.

xx
Any vector of mode numeric or character.
Details

These functions implements different input methods. In order to work these should appear as the
default values of parameters in the definition of the function whose widget is to be built. In this
way, the specification of the GUI input method for all the parameters can be done in a simple way
by means of the definition if the function. It is worth mentioning that functions defined in this way
can use parameters in the ordinary way if a value is provided for them. See the examples below.
In order to map a function onto a widget, mapFuncToWidget uses a tkentry that contains the
character conversion of the default value for that parameter if there exist such a value, or that
contains nothing there is no such a default value.
miniGUIdefaultEntry is the default entry widget, at the moment a simple tkentry.
miniGUIentry(x) makes the tkentry related to the parameter to contain x. This widget is included
as an example of the way widget can be added.
miniGUImenuSel(xx) uses ttkcombobox (needs Tcl version 8.5 or later) to show a menu with
text xx, a character or numerical vector.
These functions and their implementation show how new input widget can be added in a simple
way.

Value

All these functions returns an object miniGUIwidget, that is a list with at least the entry widget
that should be a function and any other detail.

The function widget builds an entry widget using tcltk functions and should return it. This func-
tion should be defined having three parameters: FRAME, STORE, VAR. In short, the first one is used by
the internal code to provide a tcltk parent frame, the second to provide a place where to save the
value of the parameter and the third one is used to save the parameter name.
The implementation details may change in the future.

Author(s)

Jorge Luis Ojeda Cabrera (<jojeda@unizar.es>).

See Also

miniGUI, makeWidgetCmd, miniGUIgetFormals, addMenusCmd, tcltk.

Examples

```
require(tcltk)
##
##  simple example
##
# ...define a function
h <- function(a=miniGUImenuSel(c(1,5,10)),
b=miniGUImenuSel(from=5,to=10,by=2),
c=miniGUIentry(4),
```
d=miniGUImenusel(c("T","F")),

e
)

(  cat("--h--");paste("h(a,b,c)=" , d*(a+b+c))
)
## building it
hmm <- makeWidgetCmd("Hay !!", h)
hmm()
##
## another example
##
## create some data(in the global environment)
##
n <- 100

d <- data.frame(x=runif(n))
d$z <- 0.5 * rnorm(n)
d$y <- 2 * d$x + d$z
## def mylm method
mylm <- lm
formals(mylm)$method <- quote(miniGUImenusel(c("qr","model.frame")))
formals(mylm)$x <- quote(miniGUImenusel(c("FALSE","TRUE")))
## add this stuff
miniGUI(evalPlugin, opFuns=list(mylm=mylm, lm=lm))
Index

*Topic misc
  makeWidgetCmd, 1
  mapFuncToWidget, 3
  miniGUI, 5
  miniGUIhelpers, 7
  miniGUIinputWidget, 8

*Topic utilities
  makeWidgetCmd, 1
  mapFuncToWidget, 3
  miniGUI, 5
  miniGUIhelpers, 7
  miniGUIinputWidget, 8

*Topic
  makeWidgetCmd, 1
  mapFuncToWidget, 3
  miniGUI, 5
  miniGUIhelpers, 7
  miniGUIinputWidget, 8

addMenusCmd, 2, 3, 6, 8, 9
addMenusCmd (miniGUIhelpers), 7
doNothingPlugin (miniGUIhelpers), 7
evalPlugin (miniGUIhelpers), 7
getMiniGUIans (miniGUIhelpers), 7
getMiniGUIData (miniGUIhelpers), 7
is.miniGUIwidget (miniGUIinputWidget), 8
makeWidgetCmd, 1, 2, 3, 6, 8, 9
mapFuncToWidget, 1, 3, 9
miniGUI, 2, 3, 5, 6, 8, 9
miniGUIans (miniGUIhelpers), 7
miniGUIcallEval (miniGUIhelpers), 7
miniGUIData (miniGUIhelpers), 7
miniGUIdefaultEntry (miniGUIinputWidget), 8
miniGUIentry (miniGUIinputWidget), 8
miniGUIEnvir (miniGUIhelpers), 7

miniGUIeval (miniGUIhelpers), 7
miniGUIfff (miniGUIhelpers), 7
miniGUIgetFormals, 2, 3, 9
miniGUIgetFormals (miniGUIhelpers), 7
miniGUIhelpers, 7
miniGUIinputWidget, 8
miniGUImenusel (miniGUIinputWidget), 8
miniGUIDooutput (miniGUIhelpers), 7
miniGUIscale (miniGUIinputWidget), 8

setMiniGUIans (miniGUIhelpers), 7
setMiniGUIData (miniGUIhelpers), 7
tcltk, 2, 3, 5–9
tkentry, 9
tkscale, 9
ttkcombobox, 9

11