Package ‘svDialogs’

February 20, 2015

Type Package
Version 0.9-57
Date 2014-12-19
Title SciViews GUI API - Dialog boxes
Author Philippe Grosjean [aut, cre]
Maintainer Philippe Grosjean <phgrosjean@sciviews.org>
Depends R (>= 2.6.0), svGUI (>= 0.9-52)
SystemRequirements zenity
Description Rapidly construct dialog boxes for your GUI, including an automatic function assistant
License GPL-2
URL http://www.sciviews.org/SciViews-R
BugReports https://r-forge.r-project.org/tracker/?group_id=194
NeedsCompilation no
Repository CRAN
Date/Publication 2014-12-20 10:35:02

R topics documented:

svDialogs-package .................................................. 2
dlgDir ................................................................. 2
dlgForm .............................................................. 3
dlgInput .............................................................. 5
dlgList ............................................................... 6
dlgMessage .......................................................... 7
dlgOpen .............................................................. 8
dlgSave .............................................................. 10
menuAdd ............................................................. 11

Index 14
Description

Use native or pseudo-native dialog boxes for your GUI, including an automatic function assistant.

Details

Package: svDialogs
Type: Package
Version: 0.9-57
Date: 2014-12-19
License: GPL 2 or above, at your convenience

Author(s)

Philippe Grosjean
Maintainer: Ph. Grosjean <phgrosjean@sciviews.org>

Examples

## TODO...

dlgDir

A select directory dialog box

Description

Allows to easily select a directory.

Usage

dlgDir(default = getwd(), title, ..., gui = .GUI)

## These should not be called directly
## S3 method for class 'gui'
dlgDir(default = getwd(), title, ..., gui = .GUI)
## S3 method for class 'textCLI'
dlgDir(default = getwd(), title, ..., gui = .GUI)
## S3 method for class 'nativeGUI'
dlgDir(default = getwd(), title, ..., gui = .GUI)
Arguments

- default: the directory to start with.
- title: a title to display on top of the dialog box.
- ...: pass further arguments to methods.
- gui: the 'gui' object concerned by this dialog box.

Value

The modified 'gui' object is returned invisibly. The choosen directory, or an empty string if the cancel button was clicked is found in gui$res (see example).

Author(s)

Philippe Grosjean (<phgrosjean@sciviews.org>)

See Also

dlgOpen, dlgSave

Examples

### A quick default directory changer
```r
setwd(dlgDir(default = getwd())$res)
```

---

```
# A flexible form dialog box

dlgForm

---

Description

Fill a form in a dialog box.

Usage

```r
dlgForm(form, title = "Fill the form", message = NULL, columns = 1,
        strip.type = TRUE, ..., gui = .GUI)
```

## These should not be called directly
```r
dlgForm(form, title = "Fill the form", message = NULL, columns = 1,
        strip.type = TRUE, ..., gui = .GUI)
```

## S3 method for class 'textCLI'
```r
dlgForm(form, title = "Fill the form", message = NULL, columns = 1,
        strip.type = TRUE, ..., gui = .GUI)
```

## S3 method for class 'nativeGUI'
```r
dlgForm(form, title = "Fill the form", message = NULL, columns = 1,
        strip.type = TRUE, ..., gui = .GUI)
```
Arguments

form  named list of default values, or list of possible items. Names are the labels of
      corresponding entries in the dialog box, followed by an indicator of the type of
      entry to place in the dialog box (see details).

  title  the title of the dialog box.

  message an optional message to display in the dialog box. Use \n
   for line break, or
  provide a vector of character strings, one for each line.

  columns arrange the entries on this number of columns (by row).

  strip.type do we strip the type from the names in results.

  ...  pass further arguments to methods.

  gui  the 'gui' object concerned by this dialog box.

Details

The form content is defined by a named list. Items are default values, or a list of possible items,
  e.g., for the combobox. Names are labels displayed in front of each field in the form. Follow them
by a code that represents the type of entry you want to use:

  :TXT for simple (default) textual box, :H for hidden text (password), :RO for read-only text, :NUM
  for null of positive integers with up/down arrows, :CHK for checkbox: TRUE or FALSE, :CB for read-
  only combobox, :CBE for editable combobox, :FL to select one existing file, :MFL to select multiple
  existing files, :SFL to select or create one file, :DIR to select a directory, :CDIR to select or create a
  directory, :FN to select font and font size, :DT to enter a date, :CLR to enter a RGB color, :BTN to
  create a button that execute some code, :LBL to add a label.

For the moment, the form dialog box is only supported on Linux. You have to install yad to get
access to it. On Ubuntu, you do so by:

    sudo add-apt-repository ppa:webupd8team/y-ppa-manager
    sudo apt-get update
    sudo apt-get install yad

Value

The modified 'gui' object is returned invisibly. The items entered by the user, or an empty list if the
dialog box was cancelled can be obtained from gui$res (see example).

Author(s)

Philippe Grosjean (<phgrosjean@sciviews.org>)

See Also

dlgInput
Examples

```r
## Ask a series of items at once in a dialog box
Form <- list(
  "Name:TXT" = "John Smith",
  "Age:NUM" = 25,
  "Sex:CB" = c("male", "female"),
  "Married:CHK"="false"
)
dlgForm(Form, "My form")$res
```

Description

Enter one (textual) value with this input box.

Usage

```
dlgInput(message = "Enter a value", default = "", ..., gui = .GUI)
```

Arguments

- `message` the message to display in the dialog box. Use `\n` for line break, or provide a vector of character strings, one for each line.
- `default` the default value in the text box. Single string or `NULL`.
- `...` pass further arguments to methods.
- `gui` the `gui` object concerned by this dialog box.

Value

The modified `gui` object is returned invisibly. The text entered by the user at the input box, or an empty string if the dialog box was cancelled can be obtained from `gui$res` (see example).

Author(s)

Philippe Grosjean (<phgrosjean@sciviews.org>)
See Also

dlgMessage

Examples

```r
## Ask something...
user <- dlgInput("Who are you?", Sys.info("user")$res
if (!length(user)) { # The user clicked the 'cancel' button
cat("OK, you prefer to stay anonymous!\n")
} else {
  cat("Hello", user, "\n")
}
```

dlgList

A list selection dialog box

Description

Select one or several items in a list.

Usage

dlgList(choices, preselect = NULL, multiple = FALSE, title = NULL,
...,
...,
...,
gui = .GUI)

## These should not be called directly
## S3 method for class 'gui'
dlgList(choices, preselect = NULL, multiple = FALSE, title = NULL,
...,
...,
...,
gui = .GUI)

## S3 method for class 'textCLI'
dlgList(choices, preselect = NULL, multiple = FALSE, title = NULL,
...,
...,
...,
gui = .GUI)

## S3 method for class 'nativeGUI'
dlgList(choices, preselect = NULL, multiple = FALSE, title = NULL,
...,
...,
...,
gui = .GUI)

Arguments

- `choices` the list of items. It is coerced to character strings.
- `preselect` a list of preselections, or NULL (then, the first element is selected in the list).
  Preselections not in choices are tolerated (and disregarded, of course).
- `multiple` is it a multiple selection dialog box?
- `title` the title of the dialog box, or NULL.
- `...` pass further arguments to methods.
- `gui` the 'gui' object concerned by this dialog box.
Value

The modified ‘gui’ object is returned invisibly. A list with selected items, or a character vector of length 0 if the dialog box was cancelled is available from gui$res (see examples).

Author(s)

Philippe Grosjean (<phgrosjean@sciviews.org>)

Examples

```r
## Select a month
res <- dlglist(month.name, multiple = TRUE)$res
if (!length(res)) {
  cat("You cancelled the choice\n")
} else {
  cat("You selected:\n")
  print(res)
}
```

---

dlgMessage  Display a message box

Description

A message box with icon, text, and one to three buttons.

Usage

dlgMessage(message, type = c("ok", "okcancel", "yesno", "yesnocancel"),
...,
  gui = .GUI)
msgBox(message)
okCancelBox(message)

## These should not be called directly
## S3 method for class 'gui'
dlgMessage(message, type = c("ok", "okcancel", "yesno", "yesnocancel"),
...,
  gui = .GUI)
## S3 method for class 'textCLI'
dlgMessage(message, type = c("ok", "okcancel", "yesno", "yesnocancel"),
...,
  gui = .GUI)
## S3 method for class 'nativeGUI'
dlgMessage(message, type = c("ok", "okcancel", "yesno", "yesnocancel"),
...,
  gui = .GUI)
Arguments

message the message to display in the dialog box. Use \n for line break, or provide a vector of character strings, one for each line.
type the type of dialog box: 'ok', 'okcancel', 'yesno' or 'yesnocancel'.
... pass further arguments to methods.
gui the 'gui' object concerned by this dialog box.

Value

The modified 'gui' object is returned invisibly. A string with the name of the button ("ok", "cancel", "yes" or "no") that the user pressed can be obtained from gui$res (see example). msgBox() just returns the name of the button (ok), while okCancelBox() returns TRUE if ok was clicked or FALSE if cancel was clicked.

Author(s)

Philippe Grosjean (<phgrosjean@sciviews.org>)

See Also

dlgInput

Examples

### A simple information box
dlgMessage("Hello world!")$res

### Ask to continue
dlgMessage(c("This is a long task!", "Continue?"), "okcancel")$res

### Ask a question
dlgMessage("Do you like apples?", "yesno")$res

### Idem, but one can interrupt too
text <- dlgMessage("Do you like oranges?", "yesnocancel")$res
if (text == "cancel") cat("Ah, ah! You refuse to answer!
")

### Simpler version with msgBox and okCancelBox
msgBox("Information message") # Use this to interrupt script and inform user
if (okCancelBox("Continue")) cat("we continue\n") else cat("stop it!\n")

---

dlgOpen A select file(s) to open dialog box

Description

Allows to easily select one or more existings files.
**Usage**

dlgOpen/default, title, multiple = FALSE, filters = dlgFilters["All", ], ... , gui = .GUI

dlgFilters

## These should not be called directly
## S3 method for class 'gui'
dlgOpen/default, title, multiple = FALSE, filters = dlgFilters["All", ], ... , gui = .GUI

## S3 method for class 'textCLI'
dlgOpen/default, title, multiple = FALSE, filters = dlgFilters["All", ], ... , gui = .GUI

## S3 method for class 'nativeGUI'
dlgOpen/default, title, multiple = FALSE, filters = dlgFilters["All", ], ... , gui = .GUI

**Arguments**

- **default** the default file to start with (use /dir/* or /dir/*.* to start in a given directory).
- **title** a title to display on top of the dialog box.
- **multiple** is a multiple selection of files allowed?
- **filters** a specification of file filters as a nx2 matrix, or a character string with even number of items. First item is the label, second one is the filter. See dlgFilters for examples. This is currently ignored on Mac OS X, since such kind of filter is defined differently there.
- ... pass further arguments to methods.
- **gui** the 'gui' object concerned by this dialog box.

**Value**

The modified 'gui' object is returned invisibly. The chosen file(s), or an empty string if the cancel button was clicked is found in gui$res (see example).

**Author(s)**

Philippe Grosjean (<phgrosjean@sciviews.org>)

**See Also**

dlgSave, dlgDir

**Examples**

## Choose one R file
dlgOpen(title = "Select one R file", filters = dlgFilters[c("R", "All"), ])$res

## Choose several files
dlgOpen(multiple = TRUE)$res
dlgSave

A select file to save dialog box

Description

Allows to easily select one filename to save data.

Usage

dlgSave(default, title, filters = dlgFilters["All", ], ..., gui = .GUI)

## These should not be called directly
## S3 method for class 'gui'
dlgSave(default, title, filters = dlgFilters["All", ], ..., gui = .GUI)
## S3 method for class 'textCLI'
dlgSave(default, title, filters = dlgFilters["All", ], ..., gui = .GUI)
## S3 method for class 'nativeGUI'
dlgSave(default, title, filters = dlgFilters["All", ], ..., gui = .GUI)

Arguments

default the default file to start with (use /dir/* or /dir/.. to start in a given directory, but without predefined name).
title a title to display on top of the dialog box.
filters a specification of file filters as a nx2 matrix, or a character string with even number of items. First items is the label, second one is the filter. See dlgFilters for examples. This is currently ignored on Mac OS X, since such kind of filter is defined differently there.
... pass further arguments to methods.
gui the 'gui' object concerned by this dialog box.

Value

The modified 'gui' object is returned invisibly. The choosen file, or an empty string (if the cancel button was clicked or confirmation was cancelled) is placed in gui$res (see example). For existing files, confirmation is always asked!

Author(s)

Philippe Grosjean (<phgrosjean@sciviews.org>)

See Also

dlgSave, dlgFilters, dlgDir
menuAdd

Examples

## Choose one R filename to save some R script into it
dlgSave(title = "Save R script to", filters = dlgFilters[c("R", "All"), ])$res

---

**menuAdd**

*Manage custom R menus*

**Description**

Create, populate and rework custom R menus.

**Usage**

```r
menuAdd(menuname)
menuAddItem(menuname, itemname, action)
menuDel(menuname)
menuDelItem(menuname, itemname)
menuNames()
menuItems(menuname)
```

**Arguments**

- `menuname`: a character string naming a menu.
- `itemname`: a character string naming a menu item on an existing menu.
- `action`: a character string describing the action when that menu is selected, or "none" for no action. Use "enable" or "disable" to activate or desactivate an existing menu item.

**Details**

On Windows, the function manages custom menus in RGui the same way as winMenuAdd() and similar function do. Menus are added to the right and new menu entries are added to the bottom of the menu. It is currently not possible to add menus for Rterm.exe under Windows.

On Unix/Linux, under Gnome, you must install a little Gtk2 program called `ctxmenu`

, as well as a few other utilities to manage these menus. You can download corresponding files (GPL-2 license) and instructions from [http://www.sciviews.org/SciViews-R/ctxmenu.zip](http://www.sciviews.org/SciViews-R/ctxmenu.zip). The R code only creates menu configuration files in

```
~/.ctxmenu/tmp/
```

and only in interactive R session and after the user agrees to do so (unless options(svDialogs.tmpfiles = TRUE)). Once you installed these files, you can access the menus by setting up keyboard shortcuts to activate main and context menus. The respective commands are
ctxmenu-main

and

ctxmenu-context

and you can use the preference panel to assign, e.g., <shift-menu> and <ctrl-menu>, or other keyboard shortcuts to these commands. Once everything is set up, you should see your menus appearing when a console where R + `svDialogs` runs is the active window and you hit these shortcuts. Note also that you can define custom context menus for other applications too, see the README file in the ctxmenu download.

On Mac OS X, these functions are not implemented yet (but see source of the package for experimental code commented out and try the JGR version for a first implementation there).

Action is treated as R input (echoed at the command line, parsed and executed), except if it is `none`. In this case, no action is done when the menu item is selected (merely as a placeholder for future menu actions). You can change the action of an existing menu by reissuing the command with a different action argument.

If the menuname parameter of `menuAddItem()` does not exists, it is automatically created. For creating submenus, separate successive menu names with slashes. Use `"-"` as name for separation menus under Windows or Unix/Linux.

**Value**

These function return NULL invisibly. They are used for their side-effect of creating, changing, or deleting custom R menus.

**Author(s)**

Philippe Grosjean (<phgrosjean@sciviews.org>)

**See Also**

dlgForm

**Examples**

```r
## Not run:
## Add a menu and two entries
menuAdd("Test")
menuItem("Test", "List objects", "Ls()")
menuItem("Test", "List search paths", "search()")
## Add a submenu implicitly
```
menuItem("Test/Submenu", "Current working dir", "setwd()")

## Remove Test menu and submenus
menuItem("Test")

## End(Not run)
Index

*Topic **misc**
   - dlgDir, 2
   - dlgForm, 3
   - dlgInput, 5
   - dlgList, 6
   - dlgMessage, 7
   - dlgOpen, 8
   - dlgSave, 10
   - menuAdd, 11

*Topic **package**
   - svDialogs-package, 2

*Topic **utilities**
   - svDialogs-package, 2

- dlgDir, 2, 9, 10
- dlgFilters, 10
- dlgFilters (dlgOpen), 8
- dlgForm, 3, 12
- dlgInput, 4, 5, 8
- dlgList, 6
- dlgMessage, 6, 7
- dlgOpen, 3, 8
- dlgSave, 3, 9, 10, 10

- menuAdd, 11
- menuAddItem (menuAdd), 11
- menuDel (menuAdd), 11
- menuDelItem (menuAdd), 11
- menuItems (menuAdd), 11
- menuNames (menuAdd), 11
- msgBox (dlgMessage), 7

- okCancelBox (dlgMessage), 7

- svDialogs (svDialogs-package), 2
- svDialogs-package, 2