

Package ‘twiddler’

February 20, 2015

Version 0.5-0

Title Interactive manipulation of R expressions

Description Twiddler is an interactive tool that automatically creates a Tcl/Tk GUI for manipulating variables in any R expression. See the documentation of the function `twiddle` to get started.

Author Oliver Flasch, Olaf Mersmann

Maintainer Oliver Flasch <oliver.flasch@fh-koeln.de>

Depends R (>= 2.10.0), tcltk

License GPL-2

URL <https://rsymbolic.org/projects/show/twiddler>

LazyData yes

Collate 'twiddler.R'

Date

NeedsCompilation no

Repository CRAN

Date/Publication 2013-06-17 17:27:15

R topics documented:

<code>combo</code>	2
<code>entry</code>	2
<code>filer</code>	3
<code>knob</code>	4
<code>print.twiddlerControl</code>	4
<code>shortenString</code>	5
<code>toggle</code>	5
<code>twiddle</code>	6
<code>twiddle_print</code>	7
<code>unboundVariables</code>	8

Index	9
--------------	----------

combo	<i>Combobox to manipulate a string variable by offering a list of alternative expressions</i>
-------	-----------------------------------------------------------------------------------------------

Description

combo creates a combo `twiddlerControl` for selecting elements from a list of alternative expressions as substitutions for a variable.

Usage

```
combo(..., list = NULL, label = as.character(NA))
```

Arguments

...	At least two alternative R expressions to choose from.
list	A list of arguments to append to the contents of
label	The text label of the combobox.

Value

An object of class `twiddlerControl`.

See Also

[twiddle](#)

entry	<i>Text entry field to manipulate a string variable</i>
-------	---------------------------------------------------------

Description

entry creates a text entry field `twiddlerControl` for manipulating a string variable. The variable is updated on each keypress or when the input focus enters or leaves the text field. See the `eval` parameter for details.

Usage

```
entry(default = "", label = as.character(NA),  
length = 38, eval = "key")
```

Arguments

default	The default string to display in the entry, default to "".
label	The text label of the entry.
length	The length of the entry in characters, defaults to 38.
eval	When to update the string variable manipulated by this entry. Possible values for this parameter are: "key" - update on every keystroke (the default), "focus" - update when the entry loses focus, or "return" - update when the return key is pressed.

Value

An object of class twiddlerControl.

See Also

[twiddle](#)

filer	<i>File selection dialog to manipulate a string variable denoting a filename</i>
-------	----------------------------------------------------------------------------------

Description

filer creates a file selection twiddlerControl for manipulating string variables denoting filenames.

Usage

```
filer(default = "", label = as.character(NA),
      length = 32)
```

Arguments

default	The default filename for the file selection dialog.
label	The text label of the file selection dialog.
length	The length of the entry in characters, defaults to 38.

Value

An object of class twiddlerControl.

See Also

[twiddle](#)

knob *Knob to manipulate a real valued variable*

Description

knob creates a slider twiddlerControl for manipulating real valued numerical variables.

Usage

```
knob(lim = c(0, 1), res = 0.01, default = lim[1],
     label = as.character(NA),
     ticks = abs(lim[2] - lim[1])/4, indicator = TRUE,
     length = 320)
```

Arguments

lim	A vector of two numbers, defining the lower and upper limits of the slider.
res	The resolution or step size of the slider.
default	The default numeric value D for the slider. $\text{lim}[1] \leq D \leq \text{lim}[2]$ must hold.
label	The text label of the slider.
ticks	The tick interval, defaults to $\text{abs}(\text{lim}[2] - \text{lim}[1]) / 4$. Set this to 0 to suppress drawing of ticks altogether.
indicator	Whether to show the current value as an indicator above the slider's handle, defaults to TRUE.
length	The length of the slider in pixels, default to 320.

Value

A slider twiddlerControl to be used as an argument to twiddle.

See Also

[twiddle](#)

print.twiddlerControl *Display information about a twiddler control*

Description

Display information about a twiddler control

Usage

```
## S3 method for class 'twiddlerControl'
print(x, ...)
```

Arguments

x	A twiddler control
...	Ignored

Value

Invisibly returns x.

shortenString	<i>Shorten a string if it is longer than a limit</i>
---------------	------------------------------------------------------

Description

Shorten a string if it is longer than a limit

Usage

```
shortenString(text, limit, cutIndication = "...")
```

Arguments

text	The string to shorten.
limit	The length limit.
cutIndication	A string to append to the string if shortened.

Value

The shortened string.

toggle	<i>Checkbox to manipulate a logical variable</i>
--------	--------------------------------------------------

Description

toggle creates a checkbox twiddlerControl for manipulating logical variables.

Usage

```
toggle(default = FALSE, label = as.character(NA))
```

Arguments

default	The default logical value for the checkbox, TRUE means "checked", FALSE "unchecked".
label	The text label of the checkbox.

Value

An object of class `twiddlerControl`.

See Also

[twiddle](#)

twiddle

Interactive manipulation of R expressions

Description

`twiddle` automatically creates a GUI for interactively manipulating variables in an R expression `expr`. This GUI will contain a control, i.e. a slider or a checkbox, for every unbound variable in `expr`. `expr` is re-evaluated every time the user changes a control after binding the unbound variables in `expr` to the new settings of their associated controls. This allows easy interactive exploration of parameter spaces.

Usage

```
twiddle(expr, ..., eval = TRUE, auto = TRUE,
        label = NULL, envir = parent.frame(),
        enclos = if (is.list(envir) || is.pairlist(envir)) parent.frame() else baseenv())
```

Arguments

<code>expr</code>	The expression to manipulate through GUI controls. A control is automatically created for every variable in <code>expr</code> that is not bound in the caller's environment, unless <code>auto</code> is set to <code>FALSE</code> .
<code>eval</code>	If set to <code>FALSE</code> , <code>expr</code> is not evaluated automatically each time a control is operated by the user. Instead, an "eval" button is added to the GUI to trigger manual evaluation.
<code>auto</code>	If set to <code>FALSE</code> , no controls for unbound variables will be created automatically. Unbound variables in <code>expr</code> will remain unbound, unless explicitly bound in the <code>...</code> parameter.
<code>label</code>	The text label of twiddler dialog.
<code>envir</code>	The environment in which <code>expr</code> is to be evaluated. May also be <code>NULL</code> , a list, a data frame, a pairlist or an integer as specified to sys.call .
<code>enclos</code>	Relevant when <code>envir</code> is a (pair)list or a data frame. Specifies the enclosure, i.e., where R looks for objects not found in <code>envir</code> . This can be <code>NULL</code> (interpreted as the base package environment) or an environment.
<code>...</code>	Optional configuration information for the controls for unbound variables in <code>expr</code> . This information is supplied in the form <code>variable_name = twiddlerControl, ...</code> (e.g. <code>x = knob(lim = c(0, 10))</code> , <code>b = toggle(default = TRUE)</code>). If no <code>twiddlerControl</code> object is supplied for an unbound variable, a knob with range <code>[0.0, 1.0]</code> is created as a default. This behaviour can be suppressed by setting the <code>auto</code> parameter to <code>FALSE</code> .

Value

The last values of the twiddlers as a named list.

Note

If you use an expr that generates console output as a side-effect, e.g. via `print`, and you are running Windows, you might need to use `twiddle_print` to see the output in realtime.

See Also

`knob`, `combo`, `entry`, `filer`, and `toggle`

Examples

```
## Not run:
twiddle(print(x))
twiddle(print(x), eval = FALSE)
twiddle(print(a && !b), a = toggle(), b = toggle())
twiddle(curve(sin(a * x) + cos(b * x), type = "l", col = "darkblue"),
        a = knob(c(0, 0.1), 0.001),
        b = knob(c(0, 0.2), 0.001),
        auto = FALSE)
twiddle(plot(rnorm(100), type=t), t=combo("p", "l", "b"))

## End(Not run)
```

`twiddle_print`*Print, then directly flush the output buffer*

Description

This function can be used to "print in realtime" on platforms that buffer console output, like on Windows.

Usage

```
twiddle_print(...)
```

Arguments

... Passed on to `print`

unboundVariables	<i>Find the unbound variables of an R expression</i>
------------------	------------------------------------------------------

Description

Returns the *unbound variables* of an R expression `expr`. The unbound variables are the variables in `expr` that are not bound (do not exist) in the environment of the caller of `unboundVariables`.

Usage

```
unboundVariables(qexpr)
```

Arguments

<code>qexpr</code>	The (quoted) R expression to return the unbound variables for.
--------------------	----------------------------------------------------------------

Value

The unbound variables of `expr`.

Index

combo, [2](#), [7](#)

entry, [2](#), [7](#)

filer, [3](#), [7](#)

knob, [4](#), [7](#)

print, [7](#)

print.twiddlerControl, [4](#)

shortenString, [5](#)

sys.call, [6](#)

toggle, [5](#), [7](#)

twiddle, [2-4](#), [6](#), [6](#)

twiddle_print, [7](#), [7](#)

unboundVariables, [8](#)