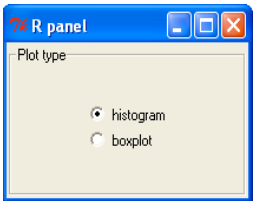
rpanel functions Quick Reference Guide

rp.control




```
rp.control(title = "", size = NULL, panelname,
realname, aschar = TRUE, ...)
```

Example

```
panel <- rp.control(x=rnorm(50),title="R panel")
rp.radiogroup(panel, plot.type, c("histogram", "boxplot"),
title="Plot type", action = hist.or.boxp)
```

rp.button

```
rp.button(panel, action, title =
deparse(substitute(action)), id = "", parent = window,
repeatdelay = 0, repeatinterval = 0, quitbutton =
FALSE, pos = NULL)
```

Example

```
boxp.sim <- function(panel) {
  boxplot(rnorm(50))
  panel
}
# Create an rpanel and add the button "Simulate" to it.
panel <- rp.control()
rp.button(panel, action = boxp.sim, title = "Simulate")
```

rp.checkbox

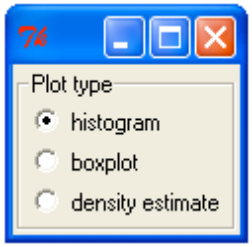
```
rp.checkbox(panel, var, action = I, labels = NA, names
= labels, title = NA, initval = NA,  parent = window,
pos = NULL, doaction = FALSE)
```

Example

```
plot.hist <- function(panel) {
  with(panel, {
    xlim <- range(c(x, mean(x) + c(-3, 3) * sd(x)))
    if (panel$cbox[3])
      clr <- "lightblue" else clr <- NULL
    hist(x, freq = FALSE, col = clr, xlim = xlim)
    if (panel$cbox[1]) {
      xgrid <- seq(xlim[1], xlim[2], length = 50)
      dgrid <- dnorm(xgrid, mean(x), sd(x))
      lines(xgrid, dgrid, col = "red", lwd = 3)
    }
    if (panel$cbox[2])
      box()
  })
  panel
}

x <- rnorm(50)
panel <- rp.control(x = x)
rp.checkbox(panel, cbox, plot.hist,
   labels = c("normal density", "box", "shading"), title =
"Options")
rp.do(panel, plot.hist)
```

rp.radiogroup




```
rp.radiogroup(panel, var, values, labels=values,
initval = values[1], parent = window, pos = NULL,
title = deparse(substitute(var)), action = I, ...)
```

Example

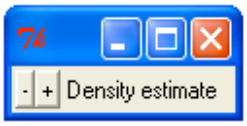
```
data.plotfn <- function(panel) {
  if (panel$plot.type == "histogram")
    hist(panel$x)
  else
    if (panel$plot.type == "boxplot")
      boxplot(panel$x)
    else
      plot(density(panel$x))
  panel
}
panel <- rp.control(x = rnorm(50))
rp.radiogroup(panel, plot.type,
       c("histogram", "boxplot", "density estimate"),
```

up

```
action = data.plotfn, title = "Plot type")
```

---

rp.doublebutton




```
rp.doublebutton(panel, var, step, title =
deparse(substitute(var)), action = I,
initval = NULL, range = c(NA, NA), log = FALSE,
showvalue = FALSE, showvaluewidth = 4, repeatinterval
= 100,
repeatdelay = 100, parent = window, pos = NULL)
```

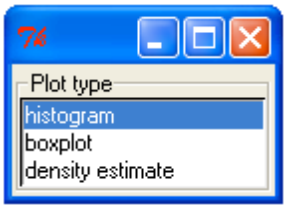
Example

```
rp.doublebutton(panel, var, step, title =
deparse(substitute(var)), action = I,

initval = NULL, range = c(NA, NA), log = FALSE, showvalue =
FALSE, showvaluewidth = 4, repeatinterval = 100,
repeatdelay = 100, parent = window, pos = NULL, row = NULL,
column = NULL, rowspan = 1, columnspan = 1, grid = NULL)
```

---

rp.listbox



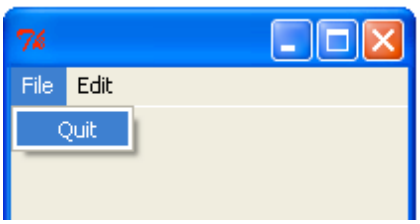

```
rp.listbox(panel, var, vals, labels = vals, rows =
length(vals), initval = vals[1],  parent = window, pos
= NULL, title = deparse(substitute(var)), action = I,
...)
```

Example

```
data.plotfn <- function(panel) {
  if (panel$plot.type == "histogram")
    hist(panel$x)
  else
    if (panel$plot.type == "boxplot")
      boxplot(panel$x)
    else
      plot(density(panel$x))
  panel
}
panel <- rp.control(x = rnorm(50))
rp.listbox(panel, plot.type,
       c("histogram", "boxplot", "density estimate"),
       action = data.plotfn, title = "Plot type")
```

---

rp.menu




```
rp.menu(panel, var, labels, initval = NULL, parent =
window, action = I, ...)
```

Example

```
a <- rp.control()
# The action function has to come first so that it already
exists for rp.menu - as it creates the callback
# functions on the fly it requires action to already be
defined.
domenu <- function(panel)
{
 rp.messagebox(panel$menuchoice, title = "You chose")
 panel
}
rp.menu(a, menuchoice,
labels=list(list("File","Quit"),list("Edit","Copy","Cut","Past
e")), action=domenu)
```

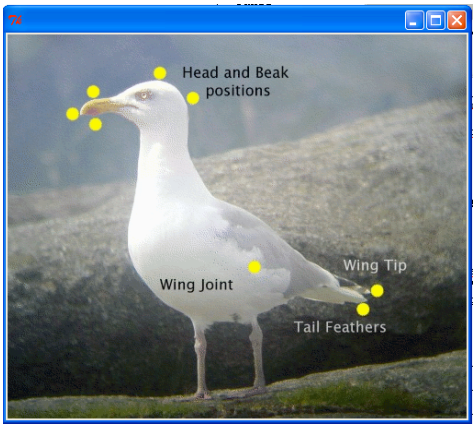
rp.slider

```
rp.slider(panel, var, from, to, action = I, title =
deparse(substitute(var)), log = FALSE, showvalue =
FALSE, resolution = 0, initval = NULL, parent =
window, pos = NULL, horizontal = TRUE)
```

Example

```
density.draw <- function(panel) {
  plot(density(panel$x, bw = panel$h))
  panel
}
panel <- rp.control(x = rnorm(50))
rp.slider(panel, h, 0.5, 5, log = TRUE, action = density.draw)
```

rp.image

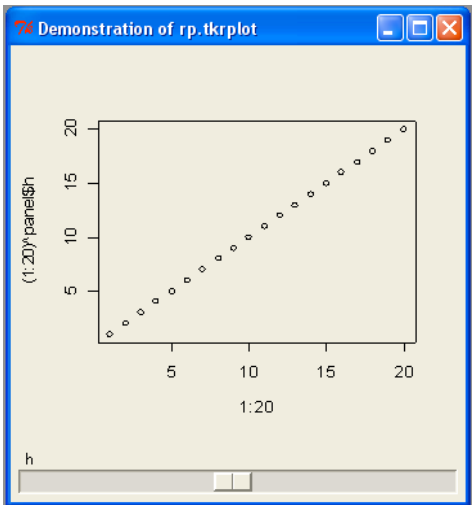



```
rp.image(panel, filename, action = NA, mousedrag = NA,
mouseup = NA, id = "img", parent = window, pos = NULL)
```

Example

```
gulls.click <- function(panel, x, y) {
  print(c(x, y))
  panel
}
panel <- rp.control()
image.file <- file.path(system.file(package = "rpanel"),
"images", "gulllmks.gif")
rp.image(panel, image.file, id = "gulls.image", action =
gulls.click)
```

rp.tkrplot




```
rp.tkrplot(panel, name, plotfun, action = NA,
mousedrag = NA, mouseup = NA, hscale = 1, vscale = 1,
parent = window, pos = NULL, foreground = NULL,
margins = c(0, 0, 0, 0))
rp.tkrreplot(panel, name)
```

Example

```
rpplot <- rp.control(title = "Demonstration of rp.tkrplot", h
= 1)

redraw <- function(panel) {
  rp.tkrreplot(panel, tkrp)
  panel
}

draw <- function(panel) {
  plot(1:20, (1:20)^panel$h)
  panel
}

rp.tkrplot(rpplot, tkrp, draw)
rp.slider(rpplot, h, action = redraw, from = 0.05, to = 2.00,
resolution = 0.05)
```