

Package ‘R.devices’

September 2, 2014

Version 2.11.0

Depends R (>= 2.10.0)

Imports R.methodsS3 (>= 1.6.1), R.oo (>= 1.18.0), R.utils (>= 1.33.0),base64enc (>= 0.1-2)

Suggests digest (>= 0.6.4), R.rsp (>= 0.19.0)

VignetteBuilder R.rsp

Author Henrik Bengtsson [aut, cre, cph]

Maintainer Henrik Bengtsson <henrikb@braju.com>

Date 2014-08-29

Title Unified handling of graphics devices

Description This package provides functions for creating plots and image files in a unified way regardless of output format (EPS, PDF, PNG, SVG, TIFF, WMF, etc.). Default device options as well as scales and aspect ratios are controlled in a uniform way across all device types. Switching output format requires minimal changes in code. This package is ideal for large-scale batch processing, because it will never leave open graphics devices or incomplete image files behind, even on errors or user interrupts.

License LGPL (>= 2.1)

LazyLoad TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-02 06:14:07

R topics documented:

R.devices-package	2
devDone	3
devEval	4
devGetLabel	6
devIsInteractive	6
devIsOpen	7
devList	8
devNew	9
devOff	10
devOptions	11
devSet	12
devSetLabel	13
toNNN	13
withPar	14

Index	16
--------------	-----------

R.devices-package	<i>Package R.devices</i>
-------------------	--------------------------

Description

This package provides functions for creating plots and image files in a unified way regardless of output format (EPS, PDF, PNG, SVG, TIFF, WMF, etc.). Default device options as well as scales and aspect ratios are controlled in a uniform way across all device types. Switching output format requires minimal changes in code. This package is ideal for large-scale batch processing, because it will never leave open graphics devices or incomplete image files behind, even on errors or user interrupts.

To get started

- Vignette '[R.devices overview](#)'
- `toEPS()`, `toPDF()`, `toPNG()`, ... - evaluate graphics code and atomically save plot to a figure file.

How to cite this package

To cite this package, please use:

```
@Manual{,
  title = {R.devices: Unified handling of graphics devices},
  author = {Henrik Bengtsson},
  year = {2014},
  note = {R package version 2.11.0},
}
```

License

LGPL (>= 2.1).

Author(s)

Henrik Bengtsson [aut, cre, cph].

devDone	<i>Closes zero or more open devices except screen (interactive) devices</i>
---------	---

Description

Closes zero or more open devices except screen (interactive) devices.

Usage

```
devDone(which=dev.cur(), ...)
```

Arguments

which	An index (numeric) vector or a label (character) vector .
...	Not used.

Value

Returns (invisibly) [dev.cur\(\)](#).

Author(s)

Henrik Bengtsson

See Also

[devOff\(\)](#). [dev.interactive](#).

devEval	<i>Opens a new graphics device, evaluate (graphing) code, and closes device</i>
---------	---

Description

Opens a new graphics device, evaluate (graphing) code, and closes device.

Usage

```
devEval(type=getOption("device"), expr, initially=NULL, finally=NULL,
  envir=parent.frame(), name=NULL, tags=NULL, sep=getOption("devEval/args/sep", ","),
  ..., ext=NULL, filename=NULL, path=getOption("devEval/args/path", "figures/"),
  field=getOption("devEval/args/field", NULL), onIncomplete=c("remove", "rename",
  "keep"), force=getOption("devEval/args/force", TRUE), which=dev.cur(), .exprAsIs=FALSE)
```

Arguments

type	Specifies the type of graphics device to be used. The device is created and opened using <code>devNew()</code> . Multiple types may be specified.
expr	The expression of graphing commands to be evaluated.
initially, finally	Optional expression :s to be evaluated before and after expr. If type specifies multiple devices, these optional expression :s are only evaluated ones.
envir	The environment where expr should be evaluated.
name, tags, sep	The fullname name of the image is specified as the name with optional sep-separated tags appended.
ext	The filename extension of the image file generated, if any. By default, it is inferred from argument type.
...	Additional arguments passed to <code>devNew()</code> .
filename	The filename of the image saved, if any. By default, it is composed of arguments name, tags, sep, and ext. See also below.
path	The directory where then image should be saved, if any.
field	An optional character string specifying a specific field of the named result list to be returned.
onIncomplete	A character string specifying what to do with an image file that was incompletely generated due to an interrupt or an error.
force	If <code>TRUE</code> , and the image file already exists, then it is overwritten, otherwise not.
which	A vector of devices to be copied. Only applied if argument expr is missing.
.exprAsIs	(Internal use only).

Value

Returns a [DevEvalFileProduct](#) if the device generated an image file, otherwise an [DevEvalProduct](#). If argument `field` is given, then the field of the [DevEvalProduct](#) is returned instead. *Note that the default return value may be changed in future releases.*

Generated image file

If created, the generated image file is saved in the directory specified by argument `path` with a filename consisting of the name followed by optional comma-separated tags and a filename extension given by argument `ext`.

By default, the image file is only created if the `expr` is evaluated completely. If it is, for instance, interrupted by the user or due to an error, then any incomplete/blank image file that was created will be removed. This behavior can be turned off using argument `onIncomplete`.

Author(s)

Henrik Bengtsson

See Also

To change default device parameters such as the width or the height, [devOptions\(\)](#). [devNew\(\)](#).

Examples

```
res <- devEval("png", name="MyPlot", tags=c("10", "rnd"), aspectRatio=0.7, {
  plot(1:10)
})
print(res$pathname)
# [1] "figures/MyPlot,10,rnd.png"

str(res$dataURI)
# chr "..."

res <- devEval("pdf", name="MyPlot", tags=c("10", "rnd"), aspectRatio=0.7, {
  plot(1:10)
})
print(res$pathname)
# [1] "figures/MyPlot,10,rnd.pdf"

res <- devEval("eps", name="MyPlot", tags=c("10", "rnd"), aspectRatio=0.7, {
  plot(1:10)
})
print(res$pathname)
# [1] "figures/MyPlot,10,rnd.eps"
```

devGetLabel	<i>Gets the labels of zero or more devices</i>
-------------	--

Description

Gets the labels of zero or more devices.

Usage

```
devGetLabel(which=dev.cur(), ...)
```

Arguments

which	An index (numeric) vector or a label (character) vector .
...	Not used.

Value

Returns a [character vector](#). If a device does not exist, an error is thrown.

Author(s)

Henrik Bengtsson

See Also

[devSetLabel\(\)](#) and [devList\(\)](#).

devIsInteractive	<i>Checks whether a device type is interactive or not</i>
------------------	---

Description

Checks whether a device type is interactive or not.

Usage

```
devIsInteractive(types, ...)
```

Arguments

types	A character vector .
...	Not used.

Value

Returns a [logical vector](#) with `TRUE` if the device type is interactive, otherwise `FALSE`.

Author(s)

Henrik Bengtsson

See Also

Internally, [deviceIsInteractive](#) is used.

devIsOpen	<i>Checks if zero or more devices are open or not</i>
-----------	---

Description

Checks if zero or more devices are open or not.

Usage

```
devIsOpen(which=dev.cur(), ...)
```

Arguments

which	An index (numeric vector) or a label (character vector).
...	Not used.

Value

Returns a named [logical vector](#) with `TRUE` if a device is open, otherwise `FALSE`.

Author(s)

Henrik Bengtsson

Examples

```
# -----
# Use devices for conditional processing of code.
# Close devices to rerun code.
# -----
cat("Currently opened device:\n")
print(devList())

# Alt A: Use device index counter (starting with the 16:th)
fig <- 15
if (!devIsOpen(fig <- fig + 1)) {
  devSet(fig)
}
```

```

    cat("Figure", fig, "\n")
    plot(1:10)
}
cat("Currently opened device:\n")
print(devList())

if (!devIsOpen(fig <- fig + 1)) {
  devSet(fig)
  cat("Figure", fig, "\n")
  plot(1:10)
}
cat("Currently opened device:\n")
print(devList())

# Alt B: Use device labels
if (!devIsOpen(label <- "part 1")) {
  devSet(label)
  cat("Part 1\n")
  plot(1:10)
}
cat("Currently opened device:\n")
print(devList())

if (!devIsOpen(label <- "part 2")) {
  devSet(label)
  cat("Part 2\n")
  plot(1:10)
}
cat("Currently opened device:\n")
print(devList())

```

devList

Lists the indices of the open devices named by their labels

Description

Lists the indices of the open devices named by their labels.

Usage

```
devList(interactiveOnly=FALSE, dropNull=TRUE, ...)
```

Arguments

interactiveOnly	If TRUE , only open interactive/screen devices are returned.
dropNull	If TRUE , the "null" device (device index 1) is not returned.
...	Not used.

Value

Returns a named [integer vector](#).

Author(s)

Henrik Bengtsson

See Also

[dev.list\(\)](#).

 devNew

Opens a new device

Description

Opens a new device.

Usage

```
devNew(type=getOption("device"), ..., scale=1, aspectRatio=1, par=NULL, label=NULL)
```

Arguments

type	A character string specifying the type of device to be opened. This string should match the name of an existing device function .
...	Additional arguments passed to the device function , e.g. width and height. If not given, the are inferred from devOptions() .
scale	A numeric scalar factor specifying how much the width and the height should be rescaled.
aspectRatio	A numeric ratio specifying the aspect ratio of the image. See below.
par	An optional named list of graphical settings applied, that is, passed to par , immediately after opening the device.
label	An optional character string specifying the label of the opened device.

Value

Returns the device index of the opened device.

Width and heights

The default width and height of the generated image is specific to the type of device used. There is not straightforward programatical way to infer these defaults; here we use [devOptions\(\)](#), which in most cases returns the correct defaults.

Aspect ratio

The aspect ratio of an image is the height relative to the width. If argument `height` is not given (or `NULL`), it is calculated as `aspectRatio*width` as long as they are given. Likewise, if argument `width` is not given (or `NULL`), it is calculated as `width/aspectRatio` as long as they are given. If neither `width` nor `height` is given, then `width` defaults to `devOptions(type)$width`. If both `width` and `height` are given, then `aspectRatio` is ignored.

Author(s)

Henrik Bengtsson

See Also

[devDone\(\)](#) and [devOff\(\)](#). For simplified generation of image files, see [devEval\(\)](#).

devOff

Closes zero or more devices

Description

Closes zero or more devices.

Usage

```
devOff(which=dev.cur(), ...)
```

Arguments

<code>which</code>	An index (numeric) vector or a label (character) vector .
<code>...</code>	Not used.

Value

Returns [dev.cur\(\)](#).

Author(s)

Henrik Bengtsson

See Also

[devDone\(\)](#). Internally, [dev.off\(\)](#) is used.

devOptions	<i>Gets the default device options</i>
------------	--

Description

Gets the default device options as given by predefined devices options adjusted for the default arguments of the device function.

Usage

```
devOptions(type=c("bmp", "cairo_pdf", "cairo_ps", "CairoWin", "CairoX11", "eps",
  "jpeg", "jpeg2", "pdf", "pictex", "png", "png2", "postscript", "quartz", "svg",
  "tiff", "win.metafile", "windows", "x11", "X11", "xfig"), custom=TRUE, special=TRUE,
  drop=TRUE, options=list(), ..., reset=FALSE)
```

Arguments

type	A character string or a device function specifying the device to be queried. May also be a vector , for querying device options for multiple devices.
custom	If TRUE , also the default settings specific to this function is returned. For more details, see below.
special	A logical . For more details, see below.
drop	If TRUE and only one device type is queried, then a list is returned, otherwise a matrix .
options	Optional named list of settings.
...	Optional named arguments for setting new defaults. For more details, see below.
reset	If TRUE , the device options are reset to R defaults.

Details

If argument `special` is [TRUE](#), then the `'width'` and `'height'` options are adjusted according to the rules explained for argument `'paper'` in [pdf](#), [postscript](#), and [xfig](#).

Value

If `drop=TRUE` and a single device is queried, a named [list](#) is returned, otherwise a [matrix](#) is returned. If a requested device is not implemented/supported on the current system, then "empty" results are returned. If options were set, that is, if named options were specified via `...`, then the list is returned invisibly, otherwise not.

Setting new defaults

When setting device options, the `getOption("devOptions")[[type]]` option is modified. This means that for such options to be effective, any device function needs to query also such options, which for instance is done by `devNew()`.

Also, for certain devices (`eps`, `pdf`, `postscript`, `quartz`, `windows` and `x11`), builtin R device options are set.

Author(s)

Henrik Bengtsson

Examples

```
# Tabulate some of the default settings for known devices
print(devOptions()[,c("width", "height", "bg", "fg", "pointsize")])
```

devSet	<i>Activates a device</i>
--------	---------------------------

Description

Activates a device.

Usage

```
devSet(which=dev.next(), ...)
```

Arguments

which	An index (numeric) or a label (character). If neither, then a label corresponding to the checksum of which as generated by digest is used.
...	Not used.

Value

Returns what [dev.set\(\)](#) returns.

Author(s)

Henrik Bengtsson

See Also

[devOff\(\)](#) and [devDone\(\)](#). Internally, [dev.set\(\)](#) is used.

devSetLabel	<i>Sets the label of a device</i>
-------------	-----------------------------------

Description

Sets the label of a device.

Usage

```
devSetLabel(which=dev.cur(), label, ...)
```

Arguments

which	An index (numeric) or a label (character).
label	A character string.
...	Not used.

Value

Returns nothing.

Author(s)

Henrik Bengtsson

See Also

[devGetLabel\(\)](#) and [devList\(\)](#).

toNNN	<i>Methods for creating image files of a specific format</i>
-------	--

Description

Methods for creating image files of a specific format.

Usage

```
toBMP(name, ..., field=getOption("devEval/args/field", "pathname"))
toEMF(name, ..., ext="emf", field=getOption("devEval/args/field", "pathname"))
toPDF(name, ..., field=getOption("devEval/args/field", "pathname"))
toPNG(name, ..., field=getOption("devEval/args/field", "pathname"))
toSVG(name, ..., field=getOption("devEval/args/field", "pathname"))
toTIFF(name, ..., field=getOption("devEval/args/field", "pathname"))
toWMF(name, ..., ext="wmf", field=getOption("devEval/args/field", "pathname"))
```

```
toFavicon(..., name="favicon", type="png", width=32L, height=width,
           par=list(mar=c(0,0,0,0), bg="transparent"))
```

```
toDefault(name, ...)
toQuartz(name, ...)
toX11(name, ...)
toWindows(name, ...)
```

```
toCairoWin(name, ...)
toCairoX11(name, ...)
```

Arguments

name	A character string specifying the name of the image file.
...	Additional arguments passed to devEval() , e.g. tags and aspectRatio.
ext	A character string specifying the filename extension.
field	Passed to devEval() .
width, height, type, par	Passed to devEval() .

Value

Returns by default the so called *fullname* of the generated image file. The fullname of an file is the filename without the extension.

Author(s)

Henrik Bengtsson

See Also

These functions are wrappers for [devEval\(\)](#). See [devOptions\(\)](#) to change the default dimensions for a specific device type.

withPar

Evaluate an R expression with graphical parameters set temporarily

Description

Evaluate an R expression with graphical parameters set temporarily.

Usage

```
withPar(expr, ..., args=list(), envir=parent.frame())
```

Arguments

expr	The R expression to be evaluated.
...	Named options to be used.
args	(optional) Additional named options specified as a named <code>list</code> .
envir	The <code>environment</code> in which the expression should be evaluated.

Value

Returns the results of the expression evaluated.

Author(s)

Henrik Bengtsson

See Also

Internally, `eval()` is used to evaluate the expression, and `par` to set graphical parameters.

Examples

```
withPar({
  layout(1:4)

  withPar({
    plot(1:10)
    plot(10:1)
  }, pch=4)

  withPar({
    plot(1:10)
    plot(10:1)
  }, pch=0, bg="yellow")
}, mar=c(2,2,1,1))
```

Index

- *Topic **IO**
 - withPar, 14
- *Topic **device**
 - devDone, 3
 - devEval, 4
 - devGetLabel, 6
 - devIsInteractive, 6
 - devIsOpen, 7
 - devList, 8
 - devNew, 9
 - devOff, 10
 - devOptions, 11
 - devSet, 12
 - devSetLabel, 13
 - toNNN, 13
- *Topic **documentation**
 - toNNN, 13
- *Topic **package**
 - R.devices-package, 2
- *Topic **programming**
 - withPar, 14
- *Topic **utilities**
 - devDone, 3
 - devEval, 4
 - devGetLabel, 6
 - devIsInteractive, 6
 - devIsOpen, 7
 - devList, 8
 - devNew, 9
 - devOff, 10
 - devOptions, 11
 - devSet, 12
 - devSetLabel, 13
 - toNNN, 13
- asDataURI (toNNN), 13
- character, 3, 4, 6, 7, 9–14
- dev.cur, 3, 10
- dev.interactive, 3
- dev.list, 9
- dev.off, 10
- dev.set, 12
- devDone, 3, 10, 12
- devDump (devEval), 4
- devEval, 4, 10, 14
- DevEvalFileProduct, 5
- DevEvalProduct, 5
- devGetLabel, 6, 13
- deviceIsInteractive, 7
- devIsInteractive, 6
- devIsOpen, 7
- devList, 6, 8, 13
- devNew, 4, 5, 9, 11
- devOff, 3, 10, 10, 12
- devOptions, 5, 9, 11, 14
- devSet, 12
- devSetLabel, 6, 13
- digest, 12
- environment, 4, 15
- eval, 15
- expression, 4
- FALSE, 7
- function, 9, 11
- integer, 9
- list, 4, 9, 11, 15
- logical, 7, 11
- matrix, 11
- NULL, 10
- numeric, 3, 6, 7, 9, 10, 12, 13
- par, 9, 15
- pdf, 11
- postscript, 11

R.devices (R.devices-package), 2

R.devices-package, 2

toBMP (toNNN), 13

toCairoWin (toNNN), 13

toCairoX11 (toNNN), 13

toDefault (toNNN), 13

toEMF (toNNN), 13

toEPS, 2

toEPS (toNNN), 13

toFavicon (toNNN), 13

toNNN, 13

toPDF, 2

toPDF (toNNN), 13

toPNG, 2

toPNG (toNNN), 13

toQuartz (toNNN), 13

toSVG (toNNN), 13

toTIFF (toNNN), 13

toWindows (toNNN), 13

toWMF (toNNN), 13

toX11 (toNNN), 13

TRUE, 4, 7, 8, 11

vector, 3, 4, 6, 7, 9–11

withPar, 14

xfig, 11