

Package ‘rvg’

January 7, 2023

Type Package

Title R Graphics Devices for 'Office' Vector Graphics Output

Version 0.3.2

Description Vector Graphics devices for 'Microsoft PowerPoint' and 'Microsoft Excel'. Functions extending package 'officer' are provided to embed 'DrawingML' graphics into 'Microsoft PowerPoint' presentations and 'Microsoft Excel' workbooks.

SystemRequirements C++11, libpng

License GPL-3

Encoding UTF-8

Depends R (>= 3.0)

Imports grDevices, Rcpp (>= 0.12.12), officer (>= 0.5.1), gdtools (>= 0.2.1), xml2 (>= 1.0.0), rlang

LinkingTo Rcpp, gdtools

Suggests testthat, grid

URL <https://ardata-fr.github.io/officeverse/>,
<https://davidgohel.github.io/rvg/>

BugReports <https://github.com/davidgohel/rvg/issues>

RoxygenNote 7.2.3

NeedsCompilation yes

Author David Gohel [aut, cre],
Bob Rudis [ctb] (the javascript code used by function set_attr),
Francois Brunetti [ctb] (clipping algorithms)

Maintainer David Gohel <david.gohel@ardata.fr>

Repository CRAN

Date/Publication 2023-01-07 15:30:07 UTC

R topics documented:

dml	2
dml_pptx	3
dml_xlsx	4
ph_with.dml	6
xl_add_vg	6

Index	8
--------------	----------

dml	<i>Wrap plot instructions for DrawingML plotting in Powerpoint</i>
-----	--

Description

A simple wrapper to mark the plot instructions as Vector Graphics instructions. It produces an object of class 'dml' with a corresponding method [ph_with](#).

The function enable usage of any R plot with argument `code` and with ggplot objects with argument `ggobj`.

Usage

```
dml(
  code,
  ggobj = NULL,
  bg = "white",
  fonts = list(),
  pointsize = 12,
  editable = TRUE,
  ...
)
```

Arguments

<code>code</code>	plotting instructions
<code>ggobj</code>	ggplot object to print. argument <code>code</code> will be ignored if this argument is supplied.
<code>bg, fonts, pointsize, editable</code>	Parameters passed to dml_pptx
<code>...</code>	unused arguments

background color

When dealing with a ggplot object argument `bg` will have no effect because ggplot theme is specifying background color, don't forget to define the colors you want in the theme:

```
theme(
  panel.background = element_rect(fill = "#EFEFEF"),
  plot.background = element_rect(fill = "wheat"))
```

See Also[ph_with.dml](#)**Examples**

```

anyplot <- dml(code = barplot(1:5, col = 2:6), bg = "wheat")

library(officer)
doc <- read_pptx()
doc <- add_slide(doc, "Title and Content", "Office Theme")
doc <- ph_with(doc, anyplot, location = ph_location_fullsize())
fileout <- tempfile(fileext = ".pptx")
# fileout <- "vg.pptx"
print(doc, target = fileout)

```

dml_pptx

*DrawingML graphic device for Microsoft PowerPoint***Description**

Graphics devices for Microsoft PowerPoint DrawingML format.

Usage

```

dml_pptx(
  file = "Rplots.dml",
  width = 6,
  height = 6,
  offx = 1,
  offy = 1,
  bg = "white",
  fonts = list(),
  pointsize = 12,
  editable = TRUE,
  id = 1L,
  last_rel_id = 1L,
  raster_prefix = "raster_",
  standalone = TRUE
)

```

Arguments

file	the file where output will appear.
height, width	Height and width in inches.
offx, offy	top and left origin of the plot
bg	Default background color for the plot (defaults to "white").

fonts	Named list of font names to be aliased with fonts installed on your system. If unspecified, the R default families sans, serif, mono and symbol are aliased to the family returned by <code>match_family()</code> . When you use specific fonts, you will need that font installed on your system. This can be check with package <code>gdtools</code> and function <code>gdtools::font_family_exists()</code> . An example: <code>list(sans = "Roboto", serif = "Times", mono = "Courier")</code> .
pointsize	default point size.
editable	should vector graphics elements (points, text, etc.) be editable.
id	specifies a unique identifier (integer) within the slide that will contain the DrawingML instructions.
last_rel_id	specifies the last unique identifier (integer) within relationship file that will be used to reference embedded raster images if any.
raster_prefix	string value used as prefix for png files produced when raster objects are printed on the graphical device.
standalone	produce a standalone drawingml file? If FALSE, omits xml header and namespaces.

See Also

[Devices](#)

Examples

```
dml_pptx(file = tempfile())
plot(1:11, (-5:5)^2, type = "b", main = "Simple Example")
dev.off()
```

dml_xlsx

DrawingML graphic device for Microsoft Excel

Description

Graphics devices for Microsoft Excel DrawingML format.

Usage

```
dml_xlsx(
  file = "Rplots.dml",
  width = 6,
  height = 6,
  offx = 1,
  offy = 1,
  bg = "white",
  fonts = list(),
```

```

    pointsize = 12,
    editable = TRUE,
    id = 1L,
    last_rel_id = 1L,
    raster_prefix = "raster_",
    standalone = TRUE
  )

```

Arguments

<code>file</code>	the file where output will appear.
<code>height, width</code>	Height and width in inches.
<code>offx, offy</code>	top and left origin of the plot
<code>bg</code>	Default background color for the plot (defaults to "white").
<code>fonts</code>	Named list of font names to be aliased with fonts installed on your system. If unspecified, the R default families <code>sans</code> , <code>serif</code> , <code>mono</code> and <code>symbol</code> are aliased to the family returned by <code>match_family()</code> .
<code>pointsize</code>	default point size.
<code>editable</code>	should vector graphics elements (points, text, etc.) be editable.
<code>id</code>	specifies a unique identifier (integer) within the slide that will contain the DrawingML instructions.
<code>last_rel_id</code>	specifies the last unique identifier (integer) within relationship file that will be used to reference embedded raster images if any.
<code>raster_prefix</code>	string value used as prefix for png files produced when raster objects are printed on the graphical device.
<code>standalone</code>	produce a standalone drawingml file? If FALSE, omits xml header and namespaces.

See Also

[Devices](#)

Examples

```

dml_xlsx(file = tempfile())
plot(1:11, (-5:5)^2, type = "b", main = "Simple Example")
dev.off()

```

ph_with.dml *add a plot output as vector graphics into a PowerPoint object*

Description

produces a vector graphics output from R plot instructions stored in a [dml](#) object and add the result in an rpptx object produced by [read_pptx](#).

Usage

```
## S3 method for class 'dml'
ph_with(x, value, location, ...)
```

Arguments

x	a pptx device
value	dml object
location	a location for a placeholder.
...	Arguments to be passed to methods

Examples

```
anyplot <- dml(code = barplot(1:5, col = 2:6), bg = "wheat")

library(officer)
doc <- read_pptx()
doc <- add_slide(doc, "Title and Content", "Office Theme")
doc <- ph_with(doc, anyplot, location = ph_location_fullsize())

fileout <- tempfile(fileext = ".pptx")
print(doc, target = fileout)
```

xl_add_vg *add a plot output as vector graphics into an Excel object*

Description

produces a vector graphics output from R plot instructions and add the result in an Excel sheet. by [read_xlsx](#).

Usage

```
xl_add_vg(x, sheet, code, left, top, width, height, ...)
```

Arguments

<code>x</code>	an <code>rxlsx</code> object produced by <code>officer::read_xlsx</code>
<code>sheet</code>	sheet label/name
<code>code</code>	plot instructions
<code>left, top</code>	left and top origin of the plot on the slide in inches.
<code>height, width</code>	Height and width in inches.
<code>...</code>	arguments passed on to <code>dml_xlsx</code> .

Examples

```
library(officer)
my_ws <- read_xlsx()
my_ws <- xl_add_vg(my_ws,
  sheet = "Feuil1",
  code = barplot(1:5, col = 2:6), width = 6, height = 6, left = 1, top = 2
)
fileout <- tempfile(fileext = ".xlsx")
print(my_ws, target = fileout)
```

Index

* **device**

dml_pptx, 3

dml_xlsx, 4

Devices, 4, 5

dml, 2, 6

dml_pptx, 2, 3

dml_xlsx, 4, 7

match_family, 4, 5

ph_with, 2

ph_with.dml, 3, 6

read_pptx, 6

read_xlsx, 6

xl_add_vg, 6