

Package: easeling (via r-universe)

September 29, 2026

Type Package

Title Minimal 'DrawingML' Graphics Device for 'openxlsx2'

Version 0.3.1

Description A base R graphics device that writes plots directly as 'OOXML' 'DrawingML' vector shapes (ECMA-376, <https://ecma-international.org/publications-and-standards/standards/ecma-376/>), for adding editable graphics to spreadsheets with 'openxlsx2' (via its `wb_add_drawing()` function). Has no dependency on 'cairo', 'FreeType', 'fontconfig', or 'xml2'; real font metrics are used when the 'systemfonts' package is available.

URL <https://github.com/JanMarvin/easeling>

BugReports <https://github.com/JanMarvin/easeling/issues>

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.6.0)

NeedsCompilation yes

Config/roxygen2/version 8.1.0

Suggests openxlsx2, systemfonts, testthat (>= 3.0.0)

Config/testthat/edition 3

Repository <https://janmarvin.r-universe.dev>

Date/Publication 2026-09-29 08:52:24 UTC

RemoteUrl <https://github.com/JanMarvin/easeling>

RemoteRef HEAD

RemoteSha c1d8d4135f0650b7a05821dc5b1f523769c4a649

Contents

easel_dev	2
easel_size	4
easel_xml	5
font_match	5
Index	7

easel_dev	<i>Open a DrawingML graphics device for openxlsx2</i>
-----------	---

Description

Draws directly to a standalone DrawingML XML file containing ‘xdr:sp’ shapes, suitable for ‘openxlsx2::wb_add_drawing(xml = file)’. No dependency on Cairo, FreeType, fontconfig, or xml2.

Usage

```
easel_dev(  
  file = tempfile(fileext = ".xml"),  
  width = 6,  
  height = 6,  
  units = "in",  
  pointsize = 12,  
  fontname = "Calibri",  
  bg = NA,  
  symbolfamily = "",  
  underline = FALSE,  
  strikeout = FALSE,  
  dims = NULL,  
  wb = NULL,  
  sheet = 1,  
  text_voff = 0.35,  
  metrics = NULL  
)
```

Arguments

file	Path of the XML file to write. Defaults to a temp file. For a plain string instead of a file, see [easel_xml()].
width, height	Device size, in ‘units’.
units	Unit for ‘width’ and ‘height’: “in” (default), “cm”, “mm” or “px”. Pixels are 96 per inch, the value spreadsheet applications use. Ignored when ‘dims’ is given, since a cell region already fixes the size.
pointsize	Default font pointsize.

fontname	Default font typeface (matches <code>'openxlsx2::wb_add_font()'</code> 's <code>'name'</code> argument), e.g. <code>"Calibri"</code> , <code>"Arial"</code> . Used whenever R itself doesn't request a specific family — i.e. whenever a plot's own <code>'family'/'fontfamily'</code> is unset or is one of R's generic aliases (<code>"sans"</code> , <code>"serif"</code> , <code>"mono"</code> , <code>"symbol"</code> , or <code>"'"</code>). If a plot sets an actual font name (e.g. <code>'par(family = "Georgia")'</code> or <code>'theme_minimal(base_family = "Georgia")'</code>), that takes priority over this default.
bg	Background colour drawn behind the plot. The default, <code>'NA'</code> , leaves the drawing transparent so the sheet shows through.
symbolfamily	Typeface asked for on plotmath symbols. The glyphs are written as Unicode either way, so this only matters when <code>'fontname'</code> lacks them: <code>"Cambria Math"</code> is the usual choice on Windows. Empty (the default) keeps symbols in <code>'fontname'</code> .
underline, strikeout	Apply underline/strikeout to all text on this device. Unlike <code>'fontname'</code> , these have no per-call R equivalent (base graphics has no underline/strikeout concept), so they're a device-wide setting.
dims	Optionally, a cell range such as <code>"A1:G15"</code> . If given, <code>'width'</code> and <code>'height'</code> are ignored and computed from the region via <code>[easel_size()]</code> , so the plot fills that region exactly when later anchored to the same <code>'dims'</code> .
wb, sheet	Passed to <code>[easel_size()]</code> when <code>'dims'</code> is given: the <code>'openxlsx2'</code> workbook (and sheet) to read actual column widths and row heights from.
text_voff	Vertical text calibration in em: text boxes are centre-anchored, and the baseline is placed <code>'text_voff'</code> em below the box centre. The default <code>'0.35'</code> was calibrated against Excel's line layout for Calibri; LibreOffice's optimum is around 0.24, so text there sits ~0.1 em low. Increase to shift rendered text up, decrease to shift it down, if your spreadsheet application's line layout places it visibly off; see <code>'system.file("examples", "calibrate_text.R", package = "easeling")'</code> .
metrics	Font metrics used for text layout (string widths, vertical centring, margins). <code>'NULL'</code> (default): use real metrics for <code>'fontname'</code> via the <code>'systemfonts'</code> package when it is installed, otherwise the built-in Calibri-like table. <code>'FALSE'</code> : always use the built-in table. Or a list with numeric components <code>'widths'</code> , <code>'ascents'</code> , <code>'descents'</code> , each of length 95 giving em fractions for the ASCII characters 32..126. Metrics only affect what R computes - rendering is always done by the spreadsheet application with the real font - but better metrics mean legend boxes, margins, and centring are sized for the text that will actually appear. The machine writing the file need not have the font it will be opened with, and <code>'systemfonts'</code> substitutes without saying so; <code>[font_match()]</code> reports which font was measured. To get real metrics for a font this machine lacks, install a replacement built to match it (Carlito for Calibri, Liberation Sans for Arial, Liberation Serif for Times New Roman), point <code>'systemfonts'</code> at the actual file with <code>'systemfonts::register_font("Calibri", plain = "calibri.ttf")'</code> before opening the device, or pass the numbers in <code>'metrics'</code> yourself.

Value

The output file path, invisibly.

Examples

```
f <- easel_dev(width = 6, height = 4, fontname = "Georgia")
plot(1:10, (1:10)^2, type = "b")
dev.off()
```

easel_size	Compute the device size for a spreadsheet cell region
------------	---

Description

Translates a cell range like "A1:G15" into a device size in inches, so that a plot drawn at that size fills the region exactly when anchored to the same 'dims' — the dims-first workflow: pick the region, size the device from it, then plot. Because the device is opened at the final size, R lays out text, margins, and legends for that size; nothing needs to be rescaled afterwards (DrawingML cannot rescale text in groups).

Usage

```
easel_size(dims, wb = NULL, sheet = 1)
```

Arguments

dims	A cell range such as "A1:G15" (both corners inclusive).
wb	Optionally, an 'openxlsx2' workbook ('wbWorkbook') to read the actual column widths and row heights from.
sheet	Sheet index in 'wb'. Default '1'.

Details

Column widths and row heights are taken from 'wb' when given, including custom widths/heights and the sheet defaults stored in its 'sheetFormatPr'. Without 'wb', Excel's defaults are assumed (Calibri 11: 8.43 character column width = 64px, 15pt row height = 20px). The character-to-pixel conversion uses Excel's formula for a maximum digit width of 7px; workbooks with a different base font will be off by the ratio of their digit widths.

Value

Named numeric vector 'c(width = , height =)' in inches.

Examples

```
easel_size("A1:G15")
```

easel_xml

Render plotting code straight to a DrawingML string

Description

Evaluates ‘code’ on an in-memory easeling device and returns the resulting drawing as a single character string - no file is involved. The device is opened before and closed after ‘code’, also on error.

Usage

```
easel_xml(code, ...)
```

Arguments

code	Plotting code; a braced block for multiple statements. Remember that ggplot2/tmap/lattice objects only draw when printed, so wrap them in ‘print()’.
...	Passed to [easel_dev()] (everything except ‘file’).

Value

The DrawingML as a length-one character vector, ready for ‘openxlsx2::wb_add_drawing(xml =)’.

Examples

```
xml <- easel_xml(plot(1:10), width = 4, height = 3)
```

font_match

Which font is actually measured for a typeface

Description

The machine writing a drawing need not have the font the drawing will be opened with. ‘system-fonts’ substitutes without saying so, and the plot is then laid out with one font’s widths while the drawing asks the spreadsheet application for another. Often that costs nothing, because some substitutes exist to match: Carlito for Calibri, Liberation Sans for Arial, Liberation Serif for Times New Roman. Others are unrelated fonts of quite different width. The device does not check, so call this when a drawing’s layout looks wrong and you want to know which font produced it.

Usage

```
font_match(fontname)
```

Arguments

fontname Typeface the drawing will name.

Value

A list with 'requested', 'matched' and 'substituted', or 'NULL' when 'systemfonts' is not installed.

Examples

```
font_match("Calibri")
```

Index

easel_dev, [2](#)
easel_size, [4](#)
easel_xml, [5](#)

font_match, [5](#)