Package 'svglite'

July 23, 2025

Title An 'SVG' Graphics Device

Version 2.2.1

Description A graphics device for R that produces 'Scalable Vector Graphics'. 'svglite' is a fork of the older 'RSvgDevice' package.

License GPL (>= 2)

URL https://svglite.r-lib.org, https://github.com/r-lib/svglite

BugReports https://github.com/r-lib/svglite/issues

Depends R (>= 4.1)

Imports base64enc, cli, lifecycle, rlang (>= 1.1.0), systemfonts (>= 1.2.3), textshaping (>= 0.3.0)

Suggests covr, fontquiver (>= 0.2.0), htmltools, knitr, rmarkdown, testthat (>= 3.0.0), xml2 (>= 1.0.0)

LinkingTo cpp11, systemfonts, textshaping

VignetteBuilder knitr

Config/build/compilation-database true

Config/Needs/website tidyverse/tidytemplate

Config/testthat/edition 3

Config/usethis/last-upkeep 2025-04-25

Encoding UTF-8

RoxygenNote 7.3.2

SystemRequirements libpng

NeedsCompilation yes

Author Hadley Wickham [aut],
Lionel Henry [aut],
Thomas Lin Pedersen [cre, aut] (ORCID:
<https://orcid.org/0000-0002-5147-4711>),
T Jake Luciani [aut],
Matthieu Decorde [aut],
Vaudor Lise [aut],

Tony Plate [ctb] (Early line dashing code),
David Gohel [ctb] (Line dashing code and early raster code),
Yixuan Qiu [ctb] (Improved styles; polypath implementation),
Håkon Malmedal [ctb] (Opacity code),
Posit Software, PBC [cph, fnd] (ROR: <https://ror.org/03wc8by49>)

Maintainer Thomas Lin Pedersen <thomas.pedersen@posit.co>

Repository CRAN

Date/Publication 2025-05-12 18:50:02 UTC

Contents

`add_web_fonts` *Add web font imports to an already created SVG file*

Description

This function allows you to add web fonts after creation. The result is the same as using the `web_fonts` argument in `svglite()`. Only SVGs created with svglite can get web fonts added.

Usage

```
add_web_fonts(filename, web_fonts)
```

Arguments

`filename`	The svgfile(s) or `svg` object(s) (as created by `svgstring()`) to edit
`web_fonts`	A list containing web fonts to use in the SVG. The fonts will still need to be available locally on the computer running the code, but viewers of the final SVG will not need the font if specified as a web font. Web fonts can either be specified using `font_face()` or given as a single string in which case they are taken to be URL's for an `@import` directive to e.g. Google Fonts. For the latter, you can use `fonts_as_import()` to automatically generate the string, optionally embedding the font data in it. If the passed in string is not in the form of a URL or `@import` statement then it is considered a font family name and `fonts_as_import()` will be called to convert it to an import automatically, using the default arguments.

Value

Invisibly returns `filename`. If any of elements of this were inline SVGs then these have been modified to include the imports

font_face *Create a font-face specification*

Description

Webfonts in SVG and HTML can either be specified manually using the `@font-face` at-rule, or imported from e.g. Google Fonts using the `@import` at-rule. `font_face()` helps you create a valid `@font-face` block for the `web_fonts` argument in `svglite()` and `svgstring()` functions.

Usage

```
font_face(
  family,
  woff2 = NULL,
  woff = NULL,
  ttf = NULL,
  otf = NULL,
  eot = deprecated(),
  svg = deprecated(),
  local = NULL,
  weight = NULL,
  style = NULL,
  range = NULL,
  variant = NULL,
  stretch = NULL,
  feature_setting = NULL,
  variation_setting = NULL,
  embed = FALSE
)
```

Arguments

`family`	The font family name this font should respond to.
`woff2`, `woff`, `ttf`, `otf`	URLs to the font in different formats. At least one must be given. Best browser support is provided by the woff format.
`eot`, `svg`	**[Deprecated]**
`local`	One or more font names that local installations of the font may have. If a local font is found with either of the given names it will be used and no download will happen.
`weight`	An optional value for the `font-weight` descriptor
`style`	An optional value for the `font-style` descriptor
`range`	An optional value for the `unicode-range` descriptor Will give the range of unicode values that this font will support
`variant`	An optional value for the `font-variant` descriptor

`stretch`	An optional value for the `font-stretch` descriptor
`feature_setting`	An optional value for the `font-feature-settings` descriptor It is recommended to avoid using this if possible
`variation_setting`	An optional value for the `font-variation-settings` descriptor.
`embed`	Should the font data be embedded directly in the SVG

Value

A character string with the `@font-face` block.

Examples

```
font_face(
  family = "MyHelvetica",
  ttf = "MgOpenModernaBold.ttf",
  local = c("Helvetica Neue Bold", "HelveticaNeue-Bold"),
  weight = "bold"
)
```

---

svglite *An SVG Graphics Driver*

Description

This function produces graphics compliant to the current w3 svg XML standard. The driver output is currently NOT specifying a DOCTYPE DTD.

Usage

```
svglite(
  filename = "Rplot%03d.svg",
  width = 10,
  height = 8,
  bg = "white",
  pointsize = 12,
  standalone = TRUE,
  web_fonts = list(),
  id = NULL,
  fix_text_size = TRUE,
  scaling = 1,
  always_valid = FALSE,
  file,
  system_fonts = list(),
  user_fonts = list()
)
```

Arguments

filename	The file where output will appear.
height, width	Height and width in inches.
bg	Default background color for the plot (defaults to "white").
pointsize	Default point size.
standalone	Produce a standalone svg file? If FALSE, omits xml header and default namespace.
web_fonts	A list containing web fonts to use in the SVG. The fonts will still need to be available locally on the computer running the code, but viewers of the final SVG will not need the font if specified as a web font. Web fonts can either be specified using font_face() or given as a single string in which case they are taken to be URL's for an @import directive to e.g. Google Fonts. For the latter, you can use fonts_as_import() to automatically generate the string, optionally embedding the font data in it. If the passed in string is not in the form of a URL or @import statement then it is considered a font family name and fonts_as_import() will be called to convert it to an import automatically, using the default arguments.
id	A character vector of ids to assign to the generated SVG's. If creating more SVG files than supplied ids the exceeding SVG's will not have an id tag and a warning will be thrown.
fix_text_size	Should the width of strings be fixed so that it doesn't change between svg renderers depending on their font rendering? Defaults to TRUE. If TRUE each string will have the textLength CSS property set to the width calculated by systemfonts and lengthAdjust='spacingAndGlyphs'. Setting this to FALSE can be beneficial for heavy post-processing that may change content or style of strings, but may lead to inconsistencies between strings and graphic elements that depend on the dimensions of the string (e.g. label borders and background).
scaling	A scaling factor to apply to the rendered line width and text size. Useful for getting the right sizing at the dimension that you need.
always_valid	Should the svgfile be a valid svg file while it is being written to? Setting this to TRUE will incur a considerable performance hit (>50% additional rendering time) so this should only be set to TRUE if the file is being parsed while it is still being written to.
file	Identical to filename. Provided for backward compatibility.
system_fonts	**[Superseded]** *Consider using* systemfonts::register_font() *instead*. 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 font_info().
user_fonts	**[Superseded]** *Consider using* systemfonts::register_font() *instead*. Named list of fonts to be aliased with font files provided by the user rather than fonts properly installed on the system. The aliases can be fonts from the fontquiver package, strings containing a path to a font file, or a list containing name and file elements with name indicating the font alias in the SVG output and file the path to a font file.

Details

svglite provides two ways of controlling fonts: system fonts aliases and user fonts aliases. Supplying a font alias has two effects. First it determines the `font-family` property of all text anchors in the SVG output. Secondly, the font is used to determine the dimensions of graphical elements and has thus an influence on the overall aspect of the plots. This means that for optimal display, the font must be available on both the computer used to create the svg, and the computer used to render the svg. See the `fonts` vignette for more information.

Author(s)

This driver was written by T Jake Luciani <jakeluciani@yahoo.com> 2012: updated by Matthieu Decorde <matthieu.decorde@ens-lyon.fr>

References

W3C Scalable Vector Graphics (SVG): https://www.w3.org/Graphics/SVG/

See Also

pictex, postscript, Devices

Examples

```
# Save to file
svglite(tempfile("Rplots.svg"))
plot(1:11, (-5:5)^2, type = "b", main = "Simple Example")
dev.off()
```

`svgstring` *Access current SVG as a string.*

Description

This is a variation on `svglite` that makes it easy to access the current value as a string.

Usage

```
svgstring(
  width = 10,
  height = 8,
  bg = "white",
  pointsize = 12,
  standalone = TRUE,
  web_fonts = list(),
  id = NULL,
  fix_text_size = TRUE,
  scaling = 1,
```

```
  system_fonts = list(),
  user_fonts = list()
)
```

Arguments

height, width	Height and width in inches.
bg	Default background color for the plot (defaults to "white").
pointsize	Default point size.
standalone	Produce a standalone svg file? If FALSE, omits xml header and default namespace.
web_fonts	A list containing web fonts to use in the SVG. The fonts will still need to be available locally on the computer running the code, but viewers of the final SVG will not need the font if specified as a web font. Web fonts can either be specified using font_face() or given as a single string in which case they are taken to be URL's for an @import directive to e.g. Google Fonts. For the latter, you can use fonts_as_import() to automatically generate the string, optionally embedding the font data in it. If the passed in string is not in the form of a URL or @import statement then it is considered a font family name and fonts_as_import() will be called to convert it to an import automatically, using the default arguments.
id	A character vector of ids to assign to the generated SVG's. If creating more SVG files than supplied ids the exceeding SVG's will not have an id tag and a warning will be thrown.
fix_text_size	Should the width of strings be fixed so that it doesn't change between svg renderers depending on their font rendering? Defaults to TRUE. If TRUE each string will have the textLength CSS property set to the width calculated by systemfonts and lengthAdjust='spacingAndGlyphs'. Setting this to FALSE can be beneficial for heavy post-processing that may change content or style of strings, but may lead to inconsistencies between strings and graphic elements that depend on the dimensions of the string (e.g. label borders and background).
scaling	A scaling factor to apply to the rendered line width and text size. Useful for getting the right sizing at the dimension that you need.
system_fonts	**[Superseded]** *Consider using* systemfonts::register_font() *instead*. 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 font_info().
user_fonts	**[Superseded]** *Consider using* systemfonts::register_font() *instead*. Named list of fonts to be aliased with font files provided by the user rather than fonts properly installed on the system. The aliases can be fonts from the fontquiver package, strings containing a path to a font file, or a list containing name and file elements with name indicating the font alias in the SVG output and file the path to a font file.

Details

See svglite() documentation for information about specifying fonts.

Value

A function with no arguments: call the function to get the current value of the string.

Examples

```
s <- svgstring()
s()

plot.new()
s()
text(0.5, 0.5, "Hi!")
s()
dev.off()

s <- svgstring()
plot(rnorm(5), rnorm(5))
s()
dev.off()
```

Index