

# Package ‘svglite’

February 7, 2020

**Version** 1.2.3

**Title** An 'SVG' Graphics Device

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

**Encoding** UTF-8

**Depends** R (>= 3.0.0)

**Imports** Rcpp, gdtools (>= 0.1.6)

**LinkingTo** Rcpp, gdtools, BH

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

**License** GPL (>= 2)

**URL** <https://github.com/r-lib/svglite>

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

**RoxygenNote** 7.0.2

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Hadley Wickham [aut],  
Lionel Henry [aut],  
Thomas Lin Pedersen [cre, aut]  
(<<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 raster code),  
Yixuan Qiu [ctb] (Improved styles; polypath implementation),  
Håkon Malmedal [ctb] (Opacity code),  
RStudio [cph]

**Maintainer** Thomas Lin Pedersen <[thomas.pedersen@rstudio.com](mailto:thomas.pedersen@rstudio.com)>

**Repository** CRAN

**Date/Publication** 2020-02-07 17:20:02 UTC

## R topics documented:

|                     |          |
|---------------------|----------|
| editSVG . . . . .   | 2        |
| htmlSVG . . . . .   | 3        |
| stringSVG . . . . . | 3        |
| svglite . . . . .   | 4        |
| svgstring . . . . . | 5        |
| xmlSVG . . . . .    | 6        |
| <b>Index</b>        | <b>8</b> |

---

|         |  |
|---------|--|
| editSVG | <i>Run plotting code and open svg in OS/system default svg viewer or editor.</i> |
|---------|--|

---

### Description

This is useful primarily for testing or post-processing the SVG.

### Usage

```
editSVG(code, ..., width = NA, height = NA)
```

### Arguments

|        |  |
|--------|--|
| code   | Plotting code to execute.                              |
| ...    | Other arguments passed on to <a href="#">svglite</a> . |
| width  | Height and width in inches.                            |
| height | Height and width in inches.                            |

### Examples

```
if (interactive()) {
  editSVG(plot(1:10))
  editSVG(contour(volcano))
}
```

---

`htmlSVG`*Run plotting code and view svg in RStudio Viewer or web browser.*

---

**Description**

This is useful primarily for testing. Requires the `htmltools` package.

**Usage**

```
htmlSVG(code, ...)
```

**Arguments**

|                   |  |
|-------------------|--|
| <code>code</code> | Plotting code to execute.                              |
| <code>...</code>  | Other arguments passed on to <a href="#">svglite</a> . |

**Examples**

```
if (require("htmltools")) {  
  htmlSVG(plot(1:10))  
  htmlSVG(hist(rnorm(100)))  
}
```

---

`stringSVG`*Run plotting code and return svg as string*

---

**Description**

This is useful primarily for testing but can be used as an alternative to [svgstring\(\)](#).

**Usage**

```
stringSVG(code, ...)
```

**Arguments**

|                   |  |
|-------------------|--|
| <code>code</code> | Plotting code to execute.                              |
| <code>...</code>  | Other arguments passed on to <a href="#">svglite</a> . |

**Examples**

```
stringSVG(plot(1:10))
```

## 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(  
  file = "Rplots.svg",  
  width = 10,  
  height = 8,  
  bg = "white",  
  pointsize = 12,  
  standalone = TRUE,  
  system_fonts = list(),  
  user_fonts = list()  
)
```

## Arguments

|               |  |
|---------------|--|
| file          | 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.   |
| system_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> .  |
| user_fonts    | 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)*: <http://www.w3.org/Graphics/SVG/Overview.htm8>

**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()

# Supply system font aliases. First check the font can be located:
gdtools::match_family("Verdana")

# Then supply a list of aliases:
fonts <- list(sans = "Verdana", mono = "Times New Roman")
svglite(tempfile("Rplots.svg"), system_fonts = fonts)
plot.new()
text(0.5, 0.5, "Some text", family = "mono")
dev.off()

# See the fonts vignettes for more options to deal with fonts
```

---

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

**Arguments**

|                           |   |
|---------------------------|---|
| <code>width</code>        | Height and width in inches.   |
| <code>height</code>       | Height and width in inches.   |
| <code>bg</code>           | Default background color for the plot (defaults to "white").  |
| <code>pointsize</code>    | Default point size.   |
| <code>standalone</code>   | Produce a standalone svg file? If FALSE, omits xml header and default namespace.  |
| <code>system_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>user_fonts</code>   | 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 <code>fontquiver</code> 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()
```

---

xmlSVG

*Run plotting code and return svg*


---

**Description**

This is useful primarily for testing. Requires the `xml2` package.

**Usage**

```
xmlSVG(code, ..., standalone = FALSE, height = 7, width = 7)
```

**Arguments**

|            |  |
|------------|--|
| code       | Plotting code to execute.  |
| ...        | Other arguments passed on to <a href="#">svglite</a> .                           |
| standalone | Produce a standalone svg file? If FALSE, omits xml header and default namespace. |
| height     | Height and width in inches.  |
| width      | Height and width in inches.  |

**Value**

A `xml2::xml_document` object.

**Examples**

```
if (require("xml2")) {  
  x <- xmlSVG(plot(1, axes = FALSE))  
  x  
  xml_find_all(x, ".//text")  
}
```

# Index

\*Topic **device**  
svglite, 4

Devices, 5

editSVG, 2

htmlSVG, 3

match\_family, 4, 6

pictex, 5

postscript, 5

stringSVG, 3

svglite, 2, 3, 4, 5–7

svgstring, 3, 5

xmlSVG, 6