Package: unigd (via r-universe)

October 20, 2026

Type Package

Title Universal Graphics Device

Version 0.2.0

Description A unified R graphics backend. Render R graphics fast and easy to many common file formats. Provides a thread safe 'C' interface for asynchronous rendering of R graphics.

License GPL (>= 2)

Depends R (>= 4.2.0)

Imports systemfonts (>= 1.0.0)

LinkingTo cpp11 (>= 0.2.4), systemfonts

Encoding UTF-8

SystemRequirements Cairo (optional, for raster output), libtiff (optional, for TIFF output), libpng, zlib

RoxygenNote 7.3.3

URL https://github.com/nx10/unigd, https://nx10.dev/unigd/

BugReports https://github.com/nx10/unigd/issues

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

Config/testthat/edition 3

Config/Needs/website tidyverse/tidytemplate

VignetteBuilder knitr

NeedsCompilation yes

Author Florian Rupprecht [aut, cre] (ORCID: <https://orcid.org/0000-0002-1795-8624>), Kun Ren [ctb], Tatsuya Shima [ctb], Jeroen Ooms [ctb] (ORCID: <https://orcid.org/0000-0002-4035-0289>), Hadley Wickham [cph] (Author of included svglite code), Lionel Henry [cph] (Author of included svglite code), Thomas Lin Pedersen [cph] (Author and creator of included svglite code), T Jake Luciani [cph] (Author of included svglite code), Matthieu Decorde [cph]

(Author of included svglite code), Vaudor Lise [cph] (Author of included svglite code), Tony Plate [cph] (Contributor to included svglite code), David Gohel [cph] (Contributor to included svglite code), Yixuan Qiu [cph] (Contributor to included svglite code), Håkon Malmedal [cph] (Contributor to included svglite code), RStudio [cph] (Copyright holder of included svglite code), Brett Robinson [cph] (Author of included belle library), Google [cph] (Copyright holder of included material design icons), Victor Zverovich [cph] (Author of included fmt library), Andrzej Krzemienski [cph] (Author of included std::experimental::optional library)

Maintainer Florian Rupprecht <floruppr@gmail.com>

Config/pak/sysreqs libcairo2-dev libpng-dev libtiff-dev zlib1g-dev

Repository https://cran.r-universe.dev

Date/Publication 2026-02-16 08:00:02 UTC

RemoteUrl https://github.com/cran/unigd

RemoteRef HEAD

RemoteSha ca86415e8a8548028d123f88ba184573c1777247

Contents

unigd-package . . . 2
ugd . . . 4
ugd_clear . . . 5
ugd_close . . . 6
ugd_id . . . 6
ugd_info . . . 7
ugd_remove . . . 8
ugd_render . . . 9
ugd_render_inline . . . 10
ugd_renderers . . . 11
ugd_save . . . 11
ugd_save_inline . . . 12
ugd_state . . . 13
ugd_test_pattern . . . 14

Index **15**

`unigd-package` *unigd: Universal graphics device*

Description

Universal graphics device

Author(s)

Maintainer: Florian Rupprecht <floruppr@gmail.com> (ORCID)

Other contributors:

- Kun Ren <mail@renkun.me> [contributor]
- Tatsuya Shima <ts1s1andn@gmail.com> [contributor]
- Jeroen Ooms <jeroen@berkeley.edu> (ORCID) [contributor]
- Hadley Wickham <hadley@rstudio.com> (Author of included svglite code) [copyright holder]
- Lionel Henry <lionel@rstudio.com> (Author of included svglite code) [copyright holder]
- Thomas Lin Pedersen <thomas.pedersen@rstudio.com> (Author and creator of included svglite code) [copyright holder]
- T Jake Luciani <jake@apache.org> (Author of included svglite code) [copyright holder]
- Matthieu Decorde <matthieu.decorde@ens-lyon.fr> (Author of included svglite code) [copyright holder]
- Vaudor Lise <lise.vaudor@ens-lyon.fr> (Author of included svglite code) [copyright holder]
- Tony Plate (Contributor to included svglite code) [copyright holder]
- David Gohel (Contributor to included svglite code) [copyright holder]
- Yixuan Qiu (Contributor to included svglite code) [copyright holder]
- Håkon Malmedal (Contributor to included svglite code) [copyright holder]
- RStudio (Copyright holder of included svglite code) [copyright holder]
- Brett Robinson (Author of included belle library) [copyright holder]
- Google (Copyright holder of included material design icons) [copyright holder]
- Victor Zverovich (Author of included fmt library) [copyright holder]
- Andrzej Krzemienski (Author of included std::experimental::optional library) [copyright holder]

See Also

Useful links:

- https://github.com/nx10/unigd
- https://nx10.dev/unigd/
- Report bugs at https://github.com/nx10/unigd/issues

ugd *A unified R graphics backend.*

Description

This function initializes a unigd graphics device.

Usage

```
ugd(
  width = getOption("unigd.width", 720),
  height = getOption("unigd.height", 576),
  bg = getOption("unigd.bg", "white"),
  pointsize = getOption("unigd.pointsize", 12),
  system_fonts = getOption("unigd.system_fonts", list()),
  user_fonts = getOption("unigd.user_fonts", list()),
  reset_par = getOption("unigd.reset_par", FALSE)
)
```

Arguments

`width`	Graphics device width (pixels).
`height`	Graphics device height (pixels).
`bg`	Background color.
`pointsize`	Graphics device point size.
`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 `systemfonts::font_info()`.
`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.
`reset_par`	If set to `TRUE`, global graphics parameters will be saved on device start and reset every time `ugd_clear()` is called (see `graphics::par()`).

Details

All font settings and descriptions are adopted from the excellent 'svglite' package.

Value

No return value, called to initialize graphics device.

Examples

```
ugd() # Initialize graphics device

# Plot something
x <- seq(0, 3 * pi, by = 0.1)
plot(x, sin(x), type = "l")

# Render plot as SVG
ugd_render(width = 600, height = 400, as = "svg")

dev.off() # alternatively: ugd_close()
```

---

ugd_clear *Clear all unigd plot pages.*

Description

This function will only work after starting a device with `ugd()`.

Usage

```
ugd_clear(which = dev.cur())
```

Arguments

`which`	Which device (ID).

Value

Whether there were any pages to remove.

Examples

```
ugd()
plot(1, 1)
hist(rnorm(100))
ugd_clear() # Clear all previous plots
hist(rnorm(100))

dev.off()
```

ugd_close *Close unigd device.*

Description

This achieves the same effect as `grDevices::dev.off()`, but will only close the device if it has the unigd type.

Usage

```
ugd_close(which = dev.cur(), all = FALSE)
```

Arguments

`which`	Which device (ID).
`all`	Should all running unigd devices be closed.

Value

Number and name of the new active device (after the specified device has been shut down).

Examples

```
ugd()
hist(rnorm(100))
ugd_close() # Equvalent to dev.off()

ugd()
ugd()
ugd()
ugd_close(all = TRUE)
```

ugd_id *Query unigd plot IDs*

Description

Query unigd graphics device static plot IDs. Available plot IDs starting from `index` will be returned. `limit` specifies the number of plots. This function will only work after starting a device with `ugd()`.

Usage

```
ugd_id(index = 0, limit = 1, which = dev.cur(), state = FALSE)
```

Arguments

index	Plot index. If this is set to 0, the last page will be selected.
limit	Limit the number of returned IDs. If this is set to a value > 1 the returned type is a list if IDs. Set to 0 for all.
which	Which device (ID).
state	Include the current device state in the returned result (see also: ugd_state()).

Value

List containing static plot IDs.

Examples

```
ugd() # Initialize graphics device

# Page 1
plot.new()
text(.5, .5, "#1")

# Page 2
plot.new()
text(.5, .5, "#2")

# Page 3
plot.new()
text(.5, .5, "#3")

third <- ugd_id() # Get ID of page 3 (last page)
second <- ugd_id(2) # Get ID of page 2
all <- ugd_id(1, limit = Inf) # Get all IDs

ugd_remove(1) # Remove page 1
ugd_render(second) # Render page 2

dev.off() # Close device
```

---

ugd_info *unigd device information.*

Description

Access general information of a unigd graphics device. This function will only work after starting a device with ugd().

Usage

```
ugd_info(which = dev.cur())
```

Arguments

which	Which device (ID).

Value

List of status variables with the following named items: $id: Server unique ID, $version: unigd and library versions.

Examples

```
ugd() # Initialize graphics device
ugd_info() # Get device information
dev.off() # Close device
```

---

ugd_remove *Remove a unigd plot page.*

Description

This function will only work after starting a device with ugd().

Usage

```
ugd_remove(page = 0, which = dev.cur())
```

Arguments

page	Plot page to remove. If this is set to 0, the last page will be selected. Can be set to a numeric plot index or plot ID (see ugd_id()).
which	Which device (ID).

Value

Whether the page existed (and thereby was successfully removed).

Examples

```
ugd()
plot(1, 1) # page 1
hist(rnorm(100)) # page 2
ugd_remove(page = 1) # remove page 1

dev.off()
```

ugd_render *Render unigd plot and return it.*

Description

See ugd_save() for saving rendered plots as files. This function will only work after starting a device with ugd().

Usage

```
ugd_render(
  page = 0,
  width = -1,
  height = -1,
  zoom = 1,
  as = "svg",
  which = dev.cur()
)
```

Arguments

page	Plot page to render. If this is set to 0, the last page will be selected. Can be set to a numeric plot index or plot ID (see ugd_id()).
width	Width of the plot. If this is set to -1, the last width will be selected.
height	Height of the plot. If this is set to -1, the last height will be selected.
zoom	Zoom level. (For example: 2 corresponds to 200%, 0.5 would be 50%.)
as	Renderer.
which	Which device (ID).

Value

Rendered plot. Text renderers return strings, binary renderers return byte arrays.

Examples

```
ugd()
plot(1, 1)
ugd_render(width = 600, height = 400, as = "svg")
dev.off()
```

`ugd_render_inline` *Inline plot rendering.*

Description

Convenience function for quick inline plot rendering. This is similar to `ugd_render()` but the plotting code is specified inline and an unigd graphics device is managed (created and closed) automatically. Starting a device with `ugd()` is therefore not necessary.

Usage

```
ugd_render_inline(
  code,
  page = 0,
  width = getOption("unigd.width", 720),
  height = getOption("unigd.height", 576),
  zoom = 1,
  as = "svg",
  ...
)
```

Arguments

`code`	Plotting code. See examples for more information.
`page`	Plot page to render. If this is set to `0`, the last page will be selected. Can be set to a numeric plot index or plot ID (see `ugd_id()`).
`width`	Width of the plot.
`height`	Height of the plot.
`zoom`	Zoom level. (For example: `2` corresponds to 200%, `0.5` would be 50%.)
`as`	Renderer.
`...`	Additional parameters passed to `ugd(...)`

Value

Rendered plot. Text renderers return strings, binary renderers return byte arrays.

Examples

```
ugd_render_inline({
  hist(rnorm(100))
}, as = "svgz")

s <- ugd_render_inline({
  plot.new()
  lines(c(0.5, 1, 0.5), c(0.5, 1, 1))
})
cat(s)
```

ugd_renderers *unigd device renderers.*

Description

Get a list of available renderers. This function will only work after starting a device with `ugd()`.

Usage

```
ugd_renderers()
```

Value

List of renderers with the following named items: `$id`: Renderer ID, `$mime`: File mime type, `$ext`: File extension, `$name`: Human readable name, `$type`: Renderer type (currently either `plot` or `other`), `$bin`: Is the file a binary blob or text.

Examples

```
ugd_renderers()
```

ugd_save *Render unigd plot to a file.*

Description

See `ugd_render()` for accessing plot data directly in memory without saving as a file. This function will only work after starting a device with `ugd()`.

Usage

```
ugd_save(
  file,
  page = 0,
  width = -1,
  height = -1,
  zoom = 1,
  as = "auto",
  which = dev.cur()
)
```

Arguments

file	Filepath to save plot.
page	Plot page to render. If this is set to 0, the last page will be selected. Can be set to a numeric plot index or plot ID (see ugd_id()).
width	Width of the plot. If this is set to -1, the last width will be selected.
height	Height of the plot. If this is set to -1, the last height will be selected.
zoom	Zoom level. (For example: 2 corresponds to 200%, 0.5 would be 50%.)
as	Renderer. When set to "auto" renderer is inferred from the file extension.
which	Which device (ID).

Value

No return value. Plot will be saved to file.

Examples

```
ugd()

plot(1, 1)

tf <- tempfile()
on.exit(unlink(tf))

ugd_save(file = tf, width = 600, height = 400, as = "png")

dev.off()
```

---

ugd_save_inline *Inline plot rendering to a file.*

Description

Convenience function for quick inline plot rendering. This is similar to ugd_save() but the plotting code is specified inline and an unigd graphics device is managed (created and closed) automatically. Starting a device with ugd() is therefore not necessary.

Usage

```
ugd_save_inline(
  code,
  file,
  page = 0,
  width = getOption("unigd.width", 720),
  height = getOption("unigd.height", 576),
  zoom = 1,
  as = "auto",
  ...
)
```

Arguments

code	Plotting code. See examples for more information.
file	Filepath to save plot.
page	Plot page to render. If this is set to 0, the last page will be selected. Can be set to a numeric plot index or plot ID (see ugd_id()).
width	Width of the plot.
height	Height of the plot.
zoom	Zoom level. (For example: 2 corresponds to 200%, 0.5 would be 50%.)
as	Renderer.
...	Additional parameters passed to ugd(...)

Value

No return value. Plot will be saved to file.

Examples

```
tf <- tempfile(fileext=".svg")
on.exit(unlink(tf))

ugd_save_inline({
  plot.new()
  lines(c(0.5, 1, 0.5), c(0.5, 1, 1))
}, file = tf)
```

---

ugd_state *unigd device status.*

Description

Access status information of a unigd graphics device. This function will only work after starting a device with ugd().

Usage

```
ugd_state(which = dev.cur())
```

Arguments

which	Which device (ID).

Value

List of status variables with the following named items: $hsize: Plot history size (how many plots are accessible), $upid: Update ID (changes when the device has received new information), $active: Is the device the currently activated device.

Examples

```
ugd()
ugd_state()
plot(1, 1)
ugd_state()

dev.off()
```

`ugd_test_pattern` *Plot a 16-panel test pattern for evaluating graphics devices.*

Description

Renders a 4x4 grid of panels, each isolating a specific drawing primitive or device feature. Covers all 8 DrawCall types (Rect, Circle, Line, Polygon, Polyline, Path, Text, Raster) plus clipping, alpha blending, line styles, text metrics, anti-aliasing hints, and point characters.

Usage

```
ugd_test_pattern()
```

Value

Invisible `NULL`; called for the side effect of drawing.

Examples

```
## Not run:

ugd_test_pattern()

## End(Not run)
```

Index

graphics::par(), *4*
grDevices::dev.off(), *6*

systemfonts::font_info(), *4*

ugd, 4
ugd(), *5–13*
ugd_clear, 5
ugd_clear(), *4*
ugd_close, 6
ugd_id, 6
ugd_id(), *8–10*, *12, 13*
ugd_info, 7
ugd_remove, 8
ugd_render, 9
ugd_render(), *10, 11*
ugd_render_inline, 10
ugd_renderers, 11
ugd_save, 11
ugd_save(), *9*, *12*
ugd_save_inline, 12
ugd_state, 13
ugd_state(), *7*
ugd_test_pattern, 14
unigd *(*unigd-package*)*, 2
unigd-package, 2