

Package: d2r (via r-universe)

October 9, 2024

Type Package

Title Create Diagrams with D2

Version 0.1.0.9000

Description Build, read, write, and render diagrams using the D2 syntax.

License MIT + file LICENSE

URL <https://github.com/elipousson/d2r>,
<https://elipousson.github.io/d2r/>

BugReports <https://github.com/elipousson/d2r/issues>

Depends R (>= 2.10)

Imports cli, fs, glue, knitr, rlang (>= 1.1.0), sys, systemfonts,
tools, utils, vctrs

Suggests magick, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

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

RemoteUrl <https://github.com/elipousson/d2r>

RemoteRef HEAD

RemoteSha 3ee2f39fde55d5a4d7e4e264008420d25e097b06

Contents

d2-utils	2
d2_arrowhead	3
d2_container	4
d2_diagram	5
d2_ggplot	6
d2_install	7
d2_read	8
d2_render	8
d2_sql_table	10
d2_style	11
is_d2_file	12
keys_d2	12
themes_d2	13

Index	14
--------------	-----------

d2-utils	<i>Basic D2 utilities</i>
----------	---------------------------

Description

Basic D2 utilities

- [d2_which\(\)](#): List path for D2
- [d2_version\(\)](#): List the installed version of D2
- [d2_themes\(\)](#): List available D2 themes
- [d2_layout\(\)](#): Lists available layout engine options with short help

Usage

`d2_which()`

`d2_version()`

`d2_themes()`

`d2_layout(layout = NULL)`

Arguments

layout	Layout name. If supplied, display long help for a particular layout engine, including its configuration options.
--------	--

d2_arrowhead	<i>Specify attributes for a single arrowhead or a pair of source and target arrowheads</i>
--------------	--

Description

`d2_arrowhead()` specifies the label, shape, and fill for a source or target arrowhead. `d2_arrowheads()` allows the specification of both a source and target arrowhead.

Usage

```
d2_arrowhead(
  shape = "triangle",
  type = c("target", "source"),
  label = NULL,
  filled = NULL
)

d2_arrowheads(..., target = list(), source = list(), id = NULL)
```

Arguments

shape	See <code>keys_d2[["arrowhead"]]</code> for allowed shape values.
type	Type of arrowhead: "target" or "source".
label	Short arrowhead label.
filled	Must be NULL or logical. Ignored if shape is "arrow" or a crow's foot shape (any shape starting with "cf-").
...	Named parameters used to set values for both the source and target arrowheads.
target, source	Named list of arguments (excluding type) passed to <code>d2_arrowhead()</code> .
id	D2 diagram map identifier. All other container or map attributes are recycled to match the length of id using <code>vecr::vec_recycle_common()</code>

Examples

```
d2_arrowhead(
  shape = "cf-many-required",
  label = "Many required"
)

example_diagram <- "A -> B"

d2_arrowheads(
  shape = "triangle",
  id = example_diagram,
  filled = FALSE
)
```

```
d2_arrowheads(
  source = list(
    shape = "triangle",
    label = "source",
    filled = FALSE
  ),
  target = list(
    shape = "circle",
    filled = TRUE
  ),
  id = example_diagram
)
```

d2_container

Create a D2 container or map

Description

A container or map is a group of diagram elements. Some special chart types, such as the SQL Table created by `d2_sql_table()` must be set up as a container. Using containers can also help set element properties including the class, label, shape and dimensions (width and height) for a set of diagram elements.

Usage

```
d2_container(
  ...,
  lines = NULL,
  id = NULL,
  class = NULL,
  name = NULL,
  label = NULL,
  shape = NULL,
  width = NULL,
  height = NULL,
  icon = NULL,
  style = d2_style(),
  collapse = NULL
)
```

Arguments

...	Diagram text.
lines	Optional diagram text. If supplied lines is recycled to match the length of id.
id	D2 diagram map identifier. All other container or map attributes are recycled to match the length of id using <code>vctrs::vec_recycle_common()</code>

class	Diagram class
name	Diagram container name
label	Diagram label
shape	Diagram shape
width, height	Diagram width and height (in pixels?)
icon	Diagram icon
style	A string or results from the <code>d2_style()</code> helper function.
collapse	If collapse = "\n", return a string with D2 diagram code for a container. If collapse = NULL (default), return a character vector with the same length as id.

Value

A string or character vector with D2 diagram code.

Examples

```
d2_container(
  "team1 <-> team2", "team2 <-> team3", "team3 <-> team1",
  id = c("round1", "round2")
)

d2_container(
  lines = c("shark -> fish: eats", "lion -> gazelle: eats"),
  id = c("ocean", "savannah")
)
```

d2_diagram

Create a D2 diagram from a named vector and connectors

Description

Note: this is an initial experimental implementation of diagram building and may change or be deprecated in the future.

Usage

```
d2_diagram(
  lines = NULL,
  ...,
  connector = NULL,
  id = NULL,
  direction = getOption("d2r.direction"),
  import = NULL,
  collapse = NULL
)
```

Arguments

lines	A list or character vector. Named elements are connected using the supplied connector value. Unnamed elements are kept as is and assumed to be valid D2 code.
...	Additional elements for diagram included after initial lines.
connector	Default to NULL which uses "->" to connect the name of any shape to the value of any shape. Must be vector of: -, ->, <-, and <->. Vector is recycled to match length of named shapes.
id	Identifier for map container for diagram.
direction	Optional direction for diagram. Must be one of: up, down, right, and left.
import	File name or names to append as an import to the top of the diagram. If named, the standard import method using the name of the import as the name of the imported element. If import is unnamed, the partial import method is used: https://d2lang.com/tour/imports#partial-imports See: https://d2lang.com/tour/imports
collapse	an optional character string to separate the results. Not <code>NA_character_</code> . When collapse is a string, the result is always a string (<code>character</code> of length 1).

Examples

```
d2_diagram(c("x" = "y"))

d2_diagram(c("a" = "b"), connector = "<-", direction = "up")

d2_diagram(c("start" = "end"), connector = "<->", import = "imported.d2")
```

d2_ggplot

Plot a D2 diagram as a ggplot2 using magick::image_ggplot()

Description

`d2_ggplot()` renders a D2 diagram input character vector or file path (to a diagram file or rendered diagram) and plots the image with `magick::image_ggplot()`.

Usage

```
d2_ggplot(
  x,
  ...,
  density = 150,
  width = NULL,
  height = NULL,
  interpolate = FALSE,
  arg = caller_arg(x),
  call = caller_env()
)
```

Arguments

x	A character vector of diagram text or a file path for a D2 diagram file to render as a diagram with <code>d2_render()</code> before plotting. A file path for a rendered D2 diagram file one using a svg, pdf, png, or gif file extension is also allowed.
...	Additional parameters passed to <code>d2_diagram()</code> .
density	Resolution to render pdf passed to <code>magick::image_read_pdf()</code> . Default 150.
width	in pixels
height	in pixels
interpolate	passed to <code>ggplot2::annotation_raster</code>
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of <code>abort()</code> for more information.

d2_install	<i>Install D2 using the sh script</i>
------------	---------------------------------------

Description

`d2_install()` downloads and executes a sh script provided for installing D2 in [the Getting Started documentation](#). This is not the most secure way to install D2. You should consider using Homebrew, a pre-built Windows installer, or another installation option listed in [the detailed installation instructions](#).

Usage

```
d2_install(dry_run = TRUE)
```

Arguments

dry_run	If TRUE (default), append "--dry-run" to the arguments passed to the sh command.
---------	--

d2_read *Read, write, and format D2 files*

Description

Read, write, and format D2 files

Usage

```
d2_read(file)

d2_write(data, file = NULL, fileext = "d2", tidy = getOption("d2r.tidy", TRUE))

d2_fmt(file)
```

Arguments

file	File name to write to disk. If NULL, set to temporary file name using supplied file extension.
data	A set of lines or a file path to a d2 file.
fileext	File extension to use for writing data to disk.
tidy	If TRUE, pass the tidy flag to the command line execution. Defaults to <code>getOption("d2r.tidy", TRUE)</code> .

d2_render *Render a D2 input file to an output file*

Description

Render a D2 input file to an output file

`d2_include()` combines `d2_render()` with `knitr::include_graphics()`.

Usage

```
d2_render(
  input,
  output = NULL,
  ...,
  fileext = "svg",
  layout = getOption("d2r.layout", "elk"),
  theme = getOption("d2r.theme"),
  sketch = getOption("d2r.sketch"),
  pad = getOption("d2r.pad"),
  animate_interval = NULL,
```



```

    font_family = NULL,
    overwrite = TRUE,
    preview = FALSE,
    call = caller_env()
  )

d2_include(
  input,
  output = NULL,
  ...,
  auto_pdf = getOption("knitr.graphics.auto_pdf", FALSE),
  dpi = NULL,
  rel_path = getOption("knitr.graphics.rel_path", TRUE),
  error = getOption("knitr.graphics.error", TRUE)
)

```

Arguments

input	Required input file with "d2" file extension.
output	Output filename.
...	Additional input flags. Optional character vector.
fileext	One of: svg, pdf, png, gif, pptx. Ignored if output is supplied.
layout	Layout. One of: elk, dagre, taga
theme	Diagram theme name or code. Theme names are not case sensitive. Use d2_themes() to list available themes.
sketch	If TRUE, use a hand-drawn style for the rendered diagram.
pad	Diagram padding in pixels. Numeric value or string coercible to whole number.
animate_interval	Required if fileext is "gif" or output uses a "gif" file extension.
font_family	String with font family name. Passed to systemfonts::match_font() to set the regular, bold, and italic fonts. Fonts must be a TTF file. D2 has a flag for setting semi-bold fonts but this is not supported at this time. font_family can also be supplied as a named vector of font family names with elements named regular, italic, and bold to use different font family names for the different styles of fonts. font_family can also be a vector of TTF file paths to allow users to provide distinct fonts for regular, italic, and bold text.
overwrite	If FALSE and output exists, abort rendering. Defaults to TRUE.
preview	If TRUE, preview the rendered plot using d2_ggplot() . Requires the magick package.
auto_pdf	Whether to use PDF images automatically when the output format is LaTeX. If TRUE, then e.g. 'foo/bar.png' will be replaced by 'foo/bar.pdf' if the latter exists. This can be useful since normally PDF images are of higher quality than raster images like PNG, when the output is LaTeX/PDF.
dpi	DPI (dots per inch) value. Used to calculate the output width (in inches) of the images. This will be their actual width in pixels, divided by dpi. If not provided, the chunk option dpi is used; if NA, the output width will not be calculated.

rel_path	Whether to automatically convert absolute paths to relative paths. If you know for sure that absolute paths work, you may set this argument or the global option <code>knitr.graphics.rel_path</code> to <code>FALSE</code> .
error	Whether to signal an error if any files specified in the path argument do not exist and are not web resources.

d2_sql_table *Create a D2 SQL Table Diagram*

Description

Create a D2 SQL Table Diagram

Usage

```
d2_sql_table(
  data = NULL,
  visibility = NULL,
  col_types = vec_ptype_abbr,
  id = NULL,
  label = id,
  ...
)
```

Arguments

data	A data frame used for column names and types. Ignored if <code>col_types</code> is a named vector. Required if <code>col_types</code> is unnamed or a function.
visibility	One or more of: <code>,</code> <code>+</code> , <code>-</code> , <code>#</code> . Length is recycled to match the number of columns in data.
col_types	A named character vector, an unnamed character vector with the same length as the number of columns in data, or a function that returns a character vector based on the columns from data. Defaults to <code>vctrs::vec_ptype_abbr</code> . Other recommended options are <code>vctrs::vec_ptype_full</code> or <code>base::typeof</code> . If types is a named character vector, the column names of data are ignored the names of types used instead.
id	Diagram container id passed to <code>d2_container()</code> . Defaults to <code>NULL</code> . If <code>NULL</code> , id is set to <code>rlang::caller_arg(data)</code> if data is supplied or a random string if <code>data = NULL</code> .
label	Diagram label. Defaults to <code>id</code> . If supplied, <code>label</code> is displayed instead of <code>id</code> .
...	Arguments passed on to <code>d2_container</code>
	lines Optional diagram text. If supplied lines is recycled to match the length of <code>id</code> .
	class Diagram class
	name Diagram container name

width,height Diagram width and height (in pixels?)
 style A string or results from the `d2_style()` helper function.
 icon Diagram icon
 collapse If collapse = "\n", return a string with D2 diagram code for a container. If collapse = NULL (default), return a character vector with the same length as id.

Value

A character string with D2 diagram code using the "sql_type" shape.

Examples

```
d2_sql_table(mtcars[, 1:4])

d2_sql_table(
  label = "col_types Example",
  col_types = c("name" = "char", "num" = "int")
)
```

d2_style	<i>Create a D2 style</i>
----------	--------------------------

Description

Learn more: <https://d2lang.com/tour/style>

Usage

```
d2_style(
  ...,
  inline = FALSE,
  level = c("shape", "connector", "root"),
  after = "\n"
)
```

Arguments

...	Named attributes for style. Underscores in parameter names are substituted for dashes to match D2 style, e.g. "stroke_dash" is converted to "stroke-dash".
inline	If TRUE or a string value, return a string instead of an array. Errors in either case and more than a single parameter is supplied. If string, the string is assumed to be the id for a shape, connector, or container/map and appended before the "style" keyword.
level	String. "shape", "connector", or "root". If "root", limit style to supported attributes for root-level style. See https://d2lang.com/tour/style#root for more information.
after	String after each line of the style specification. Defaults to "\n"

Examples

```
d2_style(stroke = "#f4a261", inline = TRUE)

d2_style(fill = "green", inline = "tree")

d2_style(animated = TRUE, stroke_dash = 3, level = "connector")
```

is_d2_file	<i>Does file use a d2 extension?</i>
------------	--------------------------------------

Description

`is_d2_file()` tests for a ".d2" file extension. A d2 file extension is not required in all cases but most functions in this package expect a d2 extension by default.

Usage

```
is_d2_file(file, fileext = "d2")
```

Arguments

file	File name(s) to test.
fileext	File extension(s) to test for.

keys_d2	<i>D2 Syntax Keywords</i>
---------	---------------------------

Description

An incomplete list of standard D2 syntax keywords for reference and internal validation. Named elements include: layout, theme, fileext, connector, arrowhead, direction, style, visibility, constraints, and grid.

Usage

```
keys_d2
```

Format

A length 10 named list.

Source

<https://d2lang.com/>

`themes_d2`*D2 Themes*

Description

A vector of standard D2 theme codes named with theme names.

Usage`themes_d2`**Format**

A length 19 double vector.

Source

<https://d2lang.com/tour/themes/>

Index

* datasets

keys_d2, [12](#)
themes_d2, [13](#)

abort(), [7](#)

base::typeof, [10](#)

character, [6](#)

d2-utils, [2](#)

d2_arrowhead, [3](#)

d2_arrowhead(), [3](#)

d2_arrowheads (d2_arrowhead), [3](#)

d2_arrowheads(), [3](#)

d2_container, [4](#), [10](#)

d2_container(), [10](#)

d2_diagram, [5](#)

d2_diagram(), [7](#)

d2_fmt (d2_read), [8](#)

d2_ggplot, [6](#)

d2_ggplot(), [6](#), [9](#)

d2_include (d2_render), [8](#)

d2_include(), [8](#)

d2_install, [7](#)

d2_install(), [7](#)

d2_layout (d2-utils), [2](#)

d2_layout(), [2](#)

d2_read, [8](#)

d2_render, [8](#)

d2_render(), [7](#), [8](#)

d2_sql_table, [10](#)

d2_sql_table(), [4](#)

d2_style, [11](#)

d2_style(), [5](#), [11](#)

d2_themes (d2-utils), [2](#)

d2_themes(), [2](#), [9](#)

d2_version (d2-utils), [2](#)

d2_version(), [2](#)

d2_which (d2-utils), [2](#)

d2_which(), [2](#)

d2_write (d2_read), [8](#)

ggplot2::annotation_raster, [7](#)

is_d2_file, [12](#)

is_d2_file(), [12](#)

keys_d2, [12](#)

knitr::include_graphics(), [8](#)

magick::image_ggplot(), [6](#)

magick::image_read_pdf(), [7](#)

NA_character_, [6](#)

systemfonts::match_font(), [9](#)

themes_d2, [13](#)

vctrs::vec_ptype_abbr, [10](#)

vctrs::vec_ptype_full, [10](#)

vctrs::vec_recycle_common(), [3](#), [4](#)