Package: grImport2 (via r-universe)

August 24, 2024

Version 0.3-3

Depends R (>= 2.12.0)

Imports methods, grDevices, grid, XML, png, jpeg, base64enc

Suggests gridSVG, rsvg

Title Importing 'SVG' Graphics

Description Functions for importing external vector images and drawing
them as part of 'R' plots. This package is different from the
'grImport' package because, where that package imports
'PostScript' format images, this package imports 'SVG' format
images. Furthermore, this package imports a specific subset of
'SVG', so external images must be preprocessed using a package
like 'rsvg' to produce 'SVG' that this package can import.
'SVG' features that are not supported by 'R' graphics, e.g.,
gradient fills, can be imported and then exported via the
'gridSVG' package.

URL https://r-forge.r-project.org/projects/grimport/,https:
//stattech.wordpress.fos.auckland.ac.nz/2013/09/,https://stattech.wordpress.
fos.auckland.ac.nz/2016/08/30/2016-11-the-butterfly-affectation/,https:
//stattech.wordpress.fos.auckland.ac.nz/2019/11/25/2019-02-svg-in-svg-out/

License GPL (>= 2)

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

RemoteUrl https://github.com/r-forge/grimport

RemoteRef HEAD

RemoteSha 7fa7711394dcb5b1c2a4381baeb93e6e06812423

Contents

---

applyTransform	*Transform a Picture Object by a Transformation Matrix*

Description

Transforms a `"Picture"` object (or indeed `"PictureContent"` objects) using a 3x3 transformation matrix.

Usage

```
applyTransform(object, tm)
```

Arguments

object	Either a `"Picture"` object or a `"PictureContent"` object.
tm	A 3x3 numeric transformation matrix.

Value

The transformed object.

Author(s)

Simon Potter

---

applyTransform-gpar   *Transform a grid* gpar *object*

Description

Transform a grid gpar object.

Methods

signature(object = "gpar", tm = "matrix") Transforms the "lwd" and "lty" components of a gpar object (if present) by a 3x3 transformation matrix and returns the modified gpar object.

Author(s)

Simon Potter

---

grid.picture   *Draw a Picture Object.*

Description

These function take a "Picture" object and either draw the picture or create a grid graphical object representing the picture.

Usage

```
pictureGrob(picture,
            x = unit(0.5, "npc"), y = unit(0.5, "npc"),
            width = NULL, height = NULL,
            just = "centre", hjust = NULL, vjust = NULL,
            default.units = "npc", expansion = 0.05,
            xscale = NULL, yscale = NULL, distort = FALSE,
            gpFUN = identity, ...,
            ext = c("none", "clipbbox", "gridSVG"),
            delayContent = match.arg(ext) == "gridSVG",
            name = NULL, prefix = NULL, clip = "on")
grid.picture(...)
```

Arguments

`picture`	A `"Picture"` object.
`x`	A single numeric value or unit object specifying an x-value.
`y`	A single numeric value or unit object specifying a y-value.
`width`	A single numeric value or unit object specifying a width. May also be `NULL` (see Details).
`height`	A single numeric value or unit object specifying a height. May also be `NULL` (see Details).
`just`	The justification of the picture relative to its (x, y) location. If there are two values, the first value specifes horizontal justification and the second value specifies vertical justification. Possible string values are: `"left"`, `"right"`, `"centre"`, `"center"`, `"bottom"`, and `"top"`. For numeric values, 0 means left alignment and 1 means right alignment.
`hjust`	A numeric vector specifying horizontal justification. If specified, overrides the `just` setting.
`vjust`	A numeric vector specifying vertical justification. If specified, overrides the `just` setting.
`default.units`	A string indicating the default units to use if `x`, `y`, `width`, or `height` are only given as numeric vectors.
`expansion`	An expansion factor; determines whether any space is left between the extent of the picture and the bounding rectangle it is drawn within.
`xscale`	A numeric vector of length two indicating the minimum and maximum on the x-scale.
`yscale`	A numeric vector of length two indicating the minimum and maximum on the y-scale.
`distort`	A logical value indicating whether the image should preserve its aspect ratio or distort to fit the area it is being drawn within.
`gpFUN`	A function that takes in a grid gpar object and returns a (optionally modified) gpar object.
`...`	For `grid.picture`, arguments to be passed to `pictureGrob`. For `pictureGrob`, arguments to be passed to the picture object's `grobify` method.
`ext`	A character vector. Selects from one of three possible extensions for drawing imported pictures. `"none"` means that no clipping will be applied to the imported picture. `"clipbbox"` means that clipping will be applied, but only to the bounding boxes of any imported clipping paths. `"gridSVG"` means that gridSVG will be used when drawing the `"Picture"` object, which enables the use of complex clipping paths, gradients, patterns, etc. to be rendered from an imported picture.
`delayContent`	If `FALSE`, `pictureGrob()` generates a gTree with children immediately. If `TRUE`, `pictureGrob()` produces a gTree with a `makeContent()` method so that children are generated only at drawing time. The latter is relevant when `ext == "gridSVG"` because it affects when SVG definitions are registered.
`name`	A character identifier.

prefix	A character string. A prefix to add to referenced gridSVG content (e.g. pattern fills). Only used when `gridSVG` is `TRUE` (for `grobify`). The reference label must be a unique reference label, otherwise an error will result. This can be checked by calling gridSVG's `listSVGDefinitions()`. When this parameter is `NULL`, a prefix will automatically be generated but this is not guaranteed to be unique.
clip	Clipping setting passed to the viewport that is set up for drawing the imported image.

Details

If `width` or `height` is `NULL`, and `distort=FALSE`, the aspect ratio of the image will be respected. By default, the image will match the width/height of the current viewport, (if `distort=FALSE`, as much as the aspect ratio allows), less any `expansion`.

Value

A grid gTree.

Author(s)

Simon Potter

See Also

[grobify](), [grid.symbols]().

---

grid.symbols	*Draw a Picture Object as Data Symbols*

Description

These functions take a `"Picture"` object and either draw the picture at several locations or create a grid graphical object representing the picture (drawn at several locations).

Usage

```
symbolsGrob(picture,
            x = stats::runif(10), y = stats::runif(10),
            size = unit(1, "char"),
            default.units = "native",
            gpFUN = identity,
            ext = c("none", "clipbbox", "gridSVG"),
            prefix = NULL,
            ..., name = NULL)
grid.symbols(...)
```

Arguments

picture	A "Picture" object.
x	A numeric vector or unit object specifying x-locations.
y	A numeric vector or unit object specifying y-locations.
size	A numeric vector or unit object specifying symbol sizes.
default.units	A string indicating the default units to use if x, y, width, or height are only given as numeric vectors.
gpFUN	A function that takes a grid gpar object and returns a (possibly modified) gpar object.
ext	A character vector. Selects from one of three possible extensions for drawing imported pictures. "none" means that no clipping will be applied to the imported picture. "clipbbox" means that clipping will be applied, but only to the bounding boxes of any imported clipping paths. "gridSVG" means that gridSVG will be used when drawing the "Picture" object, which enables the use of complex clipping paths, gradients, patterns, etc. to be rendered from an imported picture.
prefix	A character string. A prefix to add to referenced gridSVG content (e.g. pattern fills). Only used when gridSVG is TRUE. The reference label must be a unique reference label, otherwise an error will result. This can be checked by calling gridSVG's listSVGDefinitions(). When this parameter is NULL, a prefix will automatically be generated but this is not guaranteed to be unique.
...	For grid.symbols(), arguments to be passed onto symbolsGrob(). For symbolsGrob(), additional parameters to be passed onto the picture's grobify method.
name	A character identifier.

Value

A grid grob.

Author(s)

Simon Potter

See Also

[grid.picture](#)

---

grobify   *Convert a Picture Object into a grid grob*

Description

Converts "Picture" objects and pieces of a picture (i.e. objects inheriting from "PictureContent" objects) into grid grobs.

Usage

```
grobify(object, ...)
```

Arguments

object   An object that contains `grobify` methods. Typically a `"Picture"` object.

...   Further arguments to specific `grobify` methods.

Details

Although not required by some grobify methods, there are further arguments that may be used, which will be described here.

defs This argument should take a `"PictureDefinitions"` object that contains definitions of referenced content. This is only useful when the `gridSVG` is set to `TRUE`.

clip This argument is supplied in the form of a character vector. It usually only applies when grobifying a `"PictureGroup"` object. There are three possible values for `clip`:

> **none** No clipping will be applied to the object.
>
> **bbox** The clipping will be applied to the rectangular region that bounds the clipping path's content. This argument is particularly useful as R graphics can only use rectangular clipping regions. If we want more complex clipping regions, the value of `gridSVG` must be used instead.
>
> **gridSVG** When this value is set, we are no longer restricted to clipping to rectangular regions, and many images will require it to be set to display correctly. This argument requires the gridSVG package to be installed.

gridSVG This is a logical value that, when `TRUE`, allows graphical features to be drawn that are not possible in R graphics. This requires the gridSVG package. When this argument is `TRUE` we can correctly import and render patterns, masks, filters and gradients.

Value

A grid grob.

Author(s)

Simon Potter

Path Segments   *Convenience Functions for Path Segments*

Description

These functions are convenience constructors for path segment objects of classes `"PathMoveTo"`, `"PathLineTo"`, `"PathCurveTo"` and `"PathClosePath"`.

Usage

```
moveTo(x, y)
lineTo(x, y)
curveTo(x, y)
closePath(x, y)
```

Arguments

x   A numeric vector representing x-locations.

y   A numeric vector representing y-locations.

Details

For x and y, they should each be of length one so that they represent a single point location except in the case of curveTo. In the case of curveTo, it expects x and y vectors of length four, which represent control points in a cubic bezier curve.

Value

For moveTo, an object of class "PathMoveTo".

For lineTo, an object of class "PathLineTo".

For curveTo, an object of class "PathCurveTo".

For closePath, an object of class "PathClosePath".

Author(s)

Simon Potter

PathClosePath-class   *Class* "PathClosePath"

Description

A description of a path close path operator ("Z" in SVG parlance) that is meant to be used as part of a path.

Slots

x: Object of class "numeric". The x-location that the path is going to close to.

y: Object of class "numeric". The y-location that the path is going to close to.

Extends

Class PathSegment, directly.

Author(s)

Simon Potter

See Also

[PathData](#), [PathMoveTo](#), [PathLineTo](#), [PathCurveTo](#).

PathCurveTo-class   *Class* "PathCurveTo"

Description

A description of a path curve to operator ("C" in SVG parlance) that is meant to be used as part of a path.

Slots

x: Object of class "numeric". A vector of length four that represent the x-locations of the control points of a cubic bezier curve.

y: Object of class "numeric". A vector of length four that represent the y-locations of the control points of a cubic bezier curve.

Extends

Class [PathSegment](#), directly.

Author(s)

Simon Potter

See Also

[PathData](#), [PathMoveTo](#), [PathLineTo](#), [PathClosePath](#).

PathData-class   *Class* "PathData"

Description

An object that represents a sequence of path data segments in an SVG path.

Slots

segments: Object of class "list" ~~

Methods

applyTransform signature(object = "PathData", tm = "matrix"): A transformation matrix
to apply to all path segments in the path that object contains. Returns a new "PathData"
object with each path segment transformed to their new locations.

Author(s)

Simon Potter

See Also

[PathData](), [PathMoveTo](), [PathLineTo](), [PathCurveTo]().

PathLineTo-class   *Class* "PathLineTo"

Description

A description of a path line to operator ("L" in SVG parlance) that is meant to be used as part of a
path.

Slots

x: Object of class "numeric". The x-location that the path is going to draw a line to.

y: Object of class "numeric". The y-location that the path is going to draw a line to.

Extends

Class [PathSegment](), directly.

Author(s)

Simon Potter

See Also

[PathData](), [PathMoveTo](), [PathClosePath](), [PathCurveTo]().

PathMoveTo-class   *Class* "PathMoveTo"

Description

A description of a path move to operator ("M" in SVG parlance) that is meant to be used as part of a path.

Slots

x: Object of class "numeric". The x-location that the path is going to move to.

y: Object of class "numeric". The y-location that the path is going to move to.

Extends

Class [PathSegment](), directly.

Author(s)

Simon Potter

See Also

[PathData](), [PathClosePath](), [PathLineTo](), [PathCurveTo]().

PathSegment-class   *Class* "PathSegment"

Description

A segment of a path that is not intended to be used directly, merely as a convenience base class for path segment operators.

Slots

x: Object of class "numeric". An x-location.

y: Object of class "numeric". A y-location.

Methods

applyTransform signature(object = "PathSegment", tm = "matrix"): When given a 3x3 numeric transformation matrix, the points in the segment are transformed, and produce a new object representing a path segment. Not intended to be used directly on path segments, but objects that inherit from path segments (e.g. the "M", "L", "C", and "Z" operators.

Author(s)

Simon Potter

See Also

[PathMoveTo](), [PathClosePath](), [PathLineTo](), [PathCurveTo]().

Picture-class   *Class "Picture"*

Description

A collection of paths, polylines, rectangles and other graphical content and features that together describe a picture.

Objects from the Class

Objects can be created by calls of the form new("Picture", ...).

Slots

content: A list of objects of class "PictureContent" that are paths, groups rectangles and other related objects.

defs: Object of class "PictureDefinitions" that contain all referenced content in the image. These are only used by the gridSVG package when "grobifying" an image.

summary: Object of class "PictureSummary" that describes the scales applied to this picture.

Methods

[signature(x = "Picture"): subset the content of a picture to produce a new picture (including a new, updated summary).

[[signature(x = "Picture"): extract a single piece of a picture object, to produce a new picture (including a new, updated summary).

applyTransform signature(object = "Picture", tm = "matrix"): transforms each element of the picture by a 3x3 transformation matrix and returns a new "Picture" object with the newly transformed locations.

grobify signature(object = "Picture"): convert a picture into a grid grob (for use as a one-off image).

This method contains two additional arguments:

gpFUN This argument expects to take a function that modifies a gpar object. For its input, this function should take a gpar object, modify that object, and then return it. By default the value of this argument is the identity function.

ext A character vector. See [grid.picture]() for more information on what this extension selection parameter means, in addition to the valid values this argument takes.

Author(s)

Simon Potter

See Also

[readPicture](), [grid.picture](), [PictureSummary-class]().

PictureClipPath–class *Class* "PictureClipPath"

Description

A description of a clipping path that is meant to be used as part of a larger picture (as it does not directly draw anything itself). This is primarily used for determining the bounding box of the clipping region (because R graphics only supports rectangular clipping regions). This object will have a greater effect with it is used in conjunction with the gridSVG package as it can apply non-rectangular clipping regions.

Slots

content: Object of class "list". A list of "PictureContent" objects that describe the outline(s) of the clipping region(s).

label: Object of class "character". A label to identify the clipping path by so that it may be referred to by content within the image.

Extends

Class [PictureContent](), directly.

Methods

applyTransform signature(object = "PictureClipPath", tm = "matrix"): transforms the locations described by contents of the clip path object by a 3x3 transformation matrix and returns a "PictureClipPath" object whose children have had locations transformed by the matrix.

grobify signature(object = "PictureClipPath"): convert the clipping path into a grid grob. Useful only in conjunction with the gridSVG package for non-rectangular paths.

This grobify method takes three arguments:

defs Required. This argument expects a "PictureDefinitions" object that contains referenced content.

gpFUN A function that takes a gpar object and returns a modified gpar object. The default value for this arugment is the identity function.

ext A character vector. See [grid.picture]() for more information on what this extension selection parameter means, in addition to the valid values this argument takes.

Author(s)

Simon Potter

See Also

PictureGroup.

PictureContent-class   *Class* "PictureContent"

Description

A virtual class that is used for verifying validity of "Picture" objects.

Author(s)

Simon Potter

See Also

PicturePath, PictureRect, PictureGroup, etc.

PictureDefinitions-class

                   Class "PictureDefinitions"

Description

A collection of graphical content and features that are to be referenced by the rest of the image. These definitions are not intended to draw anything themselves, but describe how other pieces of graphical content will be drawn (or in the case of clipping paths, not drawn).

Slots

content: Object of class "list". A list of "PictureContent" objects.

Methods

getDef signature(id = "character"): retrieves an object that is referred to by a given label.

setDef signature(id = "character", value = "PictureContent"): Not intended to be used by regular users, but this is a method for assigning a definition to be referenced later by "PictureContent" objects. This method stores a new definition (value) with a given label (id). The result is that a new "PictureDefinitions" object is returned with the new definition added to it.

Author(s)

Simon Potter

See Also

Picture

PictureFeColorMatrix-class

Class `"PictureFeColorMatrix"`

Description

A description of a colour transformation matrix filter effect primitive. To be used in conjunction with the gridSVG package.

Slots

`type`: Object of class `"character"` ~~

`input`: Object of class `"character"` ~~

`values`: Object of class `"matrix"` ~~

`colorspace`: Object of class `"character"` ~~

Extends

Class [PictureContent](), directly.

Methods

grobify signature(object = `"PictureFeColorMatrix"`): creates an feColorMatrix object to include in a filter effect object for use with the gridSVG package.

Author(s)

Simon Potter

See Also

[PictureFilter]()

PictureFeComposite-class

Class `"PictureFeComposite"`

Description

A description of a colour transformation matrix filter effect primitive. To be used in conjunction with the gridSVG package.

Slots

input1: Object of class "character" ~~

input2: Object of class "character" ~~

operator: Object of class "character" ~~

k1: Object of class "numeric" ~~

k2: Object of class "numeric" ~~

k3: Object of class "numeric" ~~

k4: Object of class "numeric" ~~

colorspace: Object of class "character" ~~

Extends

Class [PictureContent](#), directly.

Methods

grobify signature(object = "PictureFeComposite"): creates an feComposite object to include in a filter effect object for use with the gridSVG package.

Author(s)

Paul Murrell

See Also

[PictureFilter](#)

PictureFeImage-class   *Class* "PictureFeImage"

Description

A description of a colour transformation matrix filter effect primitive. To be used in conjunction with the gridSVG package.

Slots

href: Object of class "character" ~~

result: Object of class "character" ~~

x: Object of class "numeric" ~~

y: Object of class "numeric" ~~

width: Object of class "numeric" ~~

height: Object of class "characternumeric" ~~

Extends

Class [PictureContent](), directly.

Methods

grobify signature(object = ″PictureFeImage″): creates an feImage object to include in a filter effect object for use with the gridSVG package.

Author(s)

Paul Murrell

See Also

[PictureFilter]()

PictureFilter-class   *Class* ″PictureFilter″

Description

A description of a filter effect. To be used in conjunction with the gridSVG package.

Slots

filterUnits: Object of class ″character″. The units that the filter effect should be positioned against. This should always be ″bbox″.

primitiveUnits: Object of class ″character″. The units that the filter effect should be relative to. This should always be ″coords″.

x: Object of class ″numeric″. A vector of length one representing the x-location of the bottom-left corner of the filter effect region.

y: Object of class ″numeric″. A vector of length one representing the y-location of the bottom-left corner of the filter effect region.

width: Object of class ″numeric″. A vector of length one representing the width of the filter effect region.

height: Object of class ″numeric″. A vector of length one representing the height of the filter effect region.

content: Object of class ″PictureFeColorMatrix″. A description of the colour transformation matrix applied to a filter target.

Extends

Class [PictureContent](), directly.

Methods

> **grobify** signature(object = "PictureFilter"): Creates a filter effect object for use with the gridSVG package.

Author(s)

> Simon Potter

See Also

> [PictureFeColorMatrix](#)

PictureGradientStop-class

Class "PictureGradientStop"

Description

> A description of a gradient stop that is meant to be used as part of a larger picture (as it does not directly draw anything itself). This object will have no effect unless it is used in conjunction with the gridSVG package.

Slots

> offset: Object of class "numeric". The offset (usually between 0 and 1) from the start of the gradient. These represent locations to place the colours (col) at.
>
> col: Object of class "character". A colour in the form "#RRGGBBAA".

Methods

> **grobify** signature(object = "PictureGradientStop"): convert to a gradient stop object for use with the gridSVG package.

Author(s)

> Simon Potter

See Also

> [PictureLinearGradient](#), [PictureRadialGradient](#).

PictureGroup-class   *Class* "PictureGroup"

Description

A grouped collection of picture pieces.

Slots

content: Object of class "list" that is a list of objects of class "PictureContent".

clip: Object of class "ANY". Either NULL or an object of class "PictureClipPath".

filterRef: Object of class "ANY". A character reference to an object that will apply a filter effect to this group. Not intended to be used directly (so can be NULL.

maskRef: Object of class "ANY". A character reference to an object that will mask this group. Not intended to be used directly (so can be NULL.

gp: Object of class "gpar". A grid gpar object to apply to the group.

Extends

Class "PictureContent", directly.

Methods

[signature(x = "PictureGroup"): subset the content of the group to produce a new picture.

[[signature(x = "PictureGroup"): extract a single piece of a picture group. Each piece will be an object of class "PictureContent".

applyTransform signature(object = "PictureGroup", tm = "matrix"): transforms the locations described by contents of the group object by a 3x3 transformation matrix and returns a "PictureGroup" object whose children have had locations transformed by the matrix.

grobify signature(object = "PictureGroup"): convert to a grid grob (for use as a one-off image) for drawing.

This method is typically not called directly but can take three additional arguments.

defs This argument takes an object of class "PictureDefinitions". This object allows us to make use of referenced content features in the gridSVG package.

gpFUN This argument takes a function that modifies a gpar object. The function should take a single argument that is a gpar object, modify that gpar object, and then return it. By default this argument is assigned the value of the identity function.

ext A character vector. This argument can partially match to one of off, clipbbox, or gridSVG. When off, no clipping is applied to the group. When clipbbox, the contents of the group will be clipped to the bounding box of the clipping definition. When gridSVG, the contents of the group will be clipped to the non-zero region defined by the clipping definition, in addition to using gridSVG features (i.e. gradients, filters), when required. bbox is restricted to rectangular clipping regions, while gridSVG has no such restriction and can clip to any arbitrary region.

Author(s)

Simon Potter

PictureImage-class   *Class* "PictureImage"

Description

A description of a raster image.

Slots

x: Object of class "numeric". A vector of length one representing the x-location of the top-left corner of the image.

y: Object of class "numeric". A vector of length one representing the y-location of the top-left corner of the image.

width: Object of class "numeric". A vector of length one representing the width of the image.

height: Object of class "numeric". A vector of length one representing the height of the image.

image: Object of class "nativeRaster". A description of the raster image as an array of integers representing pixel colours. Not intended to be created directly, but as a result of calling the jpeg and png packages read*() functions.

angle: Object of class "numeric". A vector of length one representing the angle applied to the image. Corresponds to grid's viewport angles.

maskRef: Object of class "ANY". A character reference to an object that will mask this image. Not intended to be used directly (so can be NULL.

bbox: Object of class "numeric". Represented as [xmin, xmax, ymin, ymax]. The bounding box of the image. Used for features such as clipping.

Extends

Class `PictureContent`, directly.

Methods

applyTransform signature(object = "PictureImage", tm = "matrix"): transforms each location described by the image by a 3x3 transformation matrix and returns a new "PictureImage" object with the newly transformed locations.

grobify signature(object = "PictureImage"): converts the image description into a grid raster grob that represents the image. Typically used in conjunction with a pattern.

While not intended to be used directly, this function contains a single optional argument, ext, which is a character vector. See `grid.picture` for more information on what this extension selection parameter means, in addition to the valid values this argument takes.

Author(s)

Simon Potter

See Also

[PictureRect](#).

PictureLinearGradient-class

Class "PictureLinearGradient"

Description

A description of a linear gradient that is meant to be used as part of a larger picture (as it does not directly draw anything itself). This object will have no effect unless it is used in conjunction with the gridSVG package.

Slots

x0: Object of class "numeric". The starting x-location of the linear gradient.

y0: Object of class "numeric". The starting y-location of the linear gradient.

x1: Object of class "numeric". The ending x-location of the linear gradient.

y1: Object of class "numeric". The ending y-location of the linear gradient.

spreadMethod: Object of class "character" that specifies what happens when a gradient ends within its bounds. Must be one of "pad", "reflect" or "repeat". See "linearGradient" in the gridSVG package for more information.

stops: Object of class "list" that is a list of objects of class "PictureGradientStop".

Extends

Class "PictureContent".

Methods

applyTransform signature(object = "PictureLinearGradient", tm = "matrix"): transform the locations that represent the bounds and direction of the gradient by a 3x3 transformation matrix.

grobify signature(object = "PictureLinearGradient"): convert to a gradient object for use with the gridSVG package.

Author(s)

Simon Potter

See Also

[Picture-class](#), [grid.picture](#).

PictureMask-class   *Class* "PictureMask"

Description

A description of a luminance mask.

Slots

content: Object of class "list". A list of "PictureContent" objects that are use to define the
mask. Typically objects that actually draw content (e.g. a path or a rect).

Extends

Class [PictureContent](#), directly.

Methods

applyTransform signature(object = "PictureMask", tm = "matrix"): transforms the loca-
tions described by contents of the mask object by a 3x3 transformation matrix and returns
a "PictureMask" object whose children have had locations transformed by the matrix.

grobify signature(object = "PictureMask"): creates a mask object for use with the gridSVG
package. Not intended to be used directly.

There are three arguments to this grobify method:

defs An object of class "PictureDefinitions". This object is primarily useful in conjunc-
tion with the gridSVG package.

gpFUN This argument takes a function that takes a gpar object and returns a modified gpar
object. The default value is the identity function.

ext A character vector. See [grid.picture](#) for more information on what this extension
selection parameter means, in addition to the valid values this argument takes.

Author(s)

Simon Potter

PicturePath-class   *Class* "PicturePath"

Description

A description of a path.

Slots

d: Object of class "PathData". A description of how the path draws and where it draws.

rule: Object of class "character". A single element vector specifying the fill rule of the path.

gp: Object of class "gpar". A grid gpar object that describes graphical parameters applied to the path.

bbox: Object of class "numeric". The bounding box of the path. Represented as [xmin, xmax, ymin, ymax]. Used for features such as clipping.

Extends

Class linkS4class{PictureContent}, directly.

Methods

applyTransform signature(object = "PicturePath", tm = "matrix"): transforms each location described by the path's data by a 3x3 transformation matrix and returns a new "PicturePath" object with the newly transformed locations.

grobify signature(object = "PicturePath"): converts the path description into a grid grob that represents the path.

Although not expected to be called directly, this method contains three arguments.

defs An object of class "PictureDefinitions". This object holds definitions of advanced graphical content, primarily for use with the gridSVG package.

gpFUN This argument expects to take a function that modifies a gpar object. This function should take a gpar object and return a modified gpar object. By default this argument is the identity function.

ext A character vector. See grid.picture for more information on what this extension selection parameter means, in addition to the valid values this argument takes.

Author(s)

Simon Potter

See Also

PictureRect, PathData.

PicturePattern-class   *Class* "PicturePattern"

Description

A description of a pattern to be referenced (and used) by graphical content.

Slots

x: Object of class "numeric". A vector of length one representing the x-location of the top-left corner of the pattern.

y: Object of class "numeric". A vector of length one representing the y-location of the top-left corner of the pattern.

width: Object of class "numeric". A vector of length one representing the width of the pattern.

height: Object of class "numeric". A vector of length one representing the height of the pattern.

angle: Object of class "numeric". A vector of length one representing the angle applied to the image. Corresponds to grid's viewport angles.

definition: Object of class "list". A list of "PictureContent" objects that define the content of the pattern "tile".

Extends

Class `PictureContent`, directly.

Methods

applyTransform signature(object = "PicturePattern", tm = "matrix"): transforms the locations described by the pattern object by a 3x3 transformation matrix and returns a "PicturePattern" object with the transformed locations.

grobify signature(object = "PicturePattern"): convert the pattern description into a gridSVG pattern object. Useful only in conjunction with the gridSVG package. The pattern object will store the definition of the pattern (almost certainly a raster of some sort) as a grob too.

While not intended to be used directly, this method contains a single optional argument. This argument is ext, a character vector. See `grid.picture` for more information on what this extension selection parameter means, in addition to the valid values this argument takes.

Author(s)

Simon Potter

PictureRadialGradient-class

Class "PictureRadialGradient"

Description

A description of a radial gradient that is meant to be used as part of a larger picture (as it does not directly draw anything itself). This object will have no effect unless it is used in conjunction with the gridSVG package.

Slots

x: Object of class "numeric". The x-location of the radial gradient.

y: Object of class "numeric". The y-location of the radial gradient.

r: Object of class "numeric". The radius of the radial gradient.

fx: Object of class "numeric". The x-location of the focal point of the radial gradient.

fy: Object of class "numeric". The y-location of the focal point of the radial gradient.

spreadMethod: Object of class "character" that specifies what happens when a gradient ends within its bounds. Must be one of "pad", "reflect" or "repeat". See "radialGradient" in the gridSVG package for more information.

stops: Object of class "list" that is a list of objects of class "PictureGradientStop".

Extends

Class "PictureContent".

Methods

applyTransform signature(object = "PictureRadialGradient", tm = "matrix"): transform the locations that represent the bounds and direction of the gradient by a 3x3 transformation matrix.

grobify signature(object = "PictureRadialGradient"): convert to a gradient object for use with the gridSVG package.

Author(s)

Simon Potter

See Also

[Picture-class](), [grid.picture]().

PictureRect-class   *Class* "PictureRect"

Description

A description of a rectangle.

Slots

> x: Object of class "numeric". A vector of length one representing the x-location of the top-left
> corner of the rectangle.
>
> y: Object of class "numeric". A vector of length one representing the y-location of the top-left
> corner of the rectangle.
>
> width: Object of class "numeric". A vector of length one representing the width of the rectangle.
>
> height: Object of class "numeric". A vector of length one representing the height of the rectangle.
>
> angle: Object of class "numeric". A vector of length one representing the angle applied to the
> image. Corresponds to grid's viewport angles.
>
> gp: Object of class "gpar". A grid gpar object.
>
> bbox: Object of class "numeric". Represented as [xmin, xmax, ymin, ymax]. The bounding box
> of the rectangle. Used for features such as clipping.

Extends

Class [PictureContent](), directly.

Methods

> **applyTransform** signature(object = "PictureRect", tm = "matrix"): transforms the loca-
> tions described by the rectangle object by a 3x3 transformation matrix and returns a "PictureRect"
> object with the transformed locations.
>
> **grobify** signature(object = "PictureRect"): converts the object into a grid rectangle grob.
>
>> While not intended to be used directly, this method contains three arguments:
>>
>> defs An object of class "PictureDefinitions" that contains definitions of graphical con-
>> tent, primarily for use with the gridSVG package.
>>
>> gpFUN This argument takes a function that should expect to take a single gpar object and
>> return a modified gpar object. By default, the value of this argument is the identity
>> function.
>>
>> ext A character vector. See [grid.picture]() for more information on what this extension
>> selection parameter means, in addition to the valid values this argument takes.

Author(s)

Simon Potter

See Also

[PicturePath]().

PictureSummary-class   *Class "PictureSummary"*

Description

Summary information about a picture regarding the scales of its bounding box.

Slots

xscale: Object of class "numeric" range of x-values in picture.

yscale: Object of class "numeric" range of y-values in picture.

Author(s)

Simon Potter

See Also

[Picture-class](#)

PictureSymbol-class   *Class "PictureSymbol"*

Description

A description of a symbol that is meant to be used (perhaps repeatedly) as part of a larger picture (as it does not directly draw anything itself). This object only has an effect while parsing a picture (via [readPicture](#)), so is not intended to be used directly by any user.

Slots

definition: Object of class "list". A list of "PictureContent" objects.

Extends

Class [PictureContent](#), directly.

Author(s)

Simon Potter

---

readPicture *Import a Cairo SVG image*

Description

This function reads in an SVG image file and creates a `"Picture"` object.

Usage

```
readPicture(file, warn = TRUE, initDefs = TRUE)
```

Arguments

`file`	The filename of a Cairo SVG image.
`warn`	Logical. If TRUE, this will warn if `file` was not generated by Cairo graphics (see Details).
`initDefs`	Logical. If TRUE, the global list of definitions (for things like masks, filters, gradient fills, ...) is initialised.

Details

This function is designed to read SVG files that have been generated by the Cairo graphics system.

There are several ways to generate Cairo SVG files: The R SVG graphics device, provided by the [svg](#) function, produces Cairo SVG; the **grConvert** package (Linux-only) can convert PostScript, or PDF, or SVG files to Cairo SVG; and the **rsvg** package can convert many different graphics formats to Cairo SVG.

It is very unlikely that an SVG file that was NOT generated by Cairo will import properly. This function may read the file without error, but the render (via [grid.picture](#)) is very unlikely to faithfully reproduce the original image.

Value

An object of class `"Picture"`.

Author(s)

Simon Potter

References

The Cairo graphics library, https://cairographics.org/

Examples

```
options(warn=1)

## NOT a Cairo SVG file
badfile <- system.file("SVG", "lwd.svg", package="grImport2")
## A Cairo SVG file
goodfile <- system.file("SVG", "lwd-rsvg.svg", package="grImport2")

## Warning because NOT a Cairo SVG file
## (and it will not render correctly)
img <- readPicture(badfile)

## No warning
## (and it will render correctly)
img <- readPicture(goodfile)

if (require("rsvg")) {
    ## Generate a Cairo SVG file
    goodfile <- tempfile(fileext = ".svg")
    rsvg_svg(badfile, goodfile)

    ## No warning
    ## (and it will render correctly)
    img <- readPicture(goodfile)

}

options(warn=0)
```

Index