

Package ‘gridGraphics’

June 6, 2017

Title Redraw Base Graphics Using 'grid' Graphics

Version 0.2

Description Functions to convert a page of plots drawn with the graphics package into identical output drawn with the grid package. The result looks like the original graphics-based plot, but consists of grid grobs and viewports that can then be manipulated with grid functions (e.g., edit grobs and revisit viewports).

Depends grid, graphics

Imports grDevices

Suggests KernSmooth

SystemRequirements ImageMagick (for testing)

License GPL (>= 2)

URL <https://github.com/pmur002/gridgraphics>

NeedsCompilation no

Author Paul Murrell [cre, aut]

Maintainer Paul Murrell <paul@stat.auckland.ac.nz>

Repository CRAN

Date/Publication 2017-06-06 11:22:27 UTC

R topics documented:

grid.echo	2
plotdiff	3

Index	5
--------------	----------

`grid.echo`*Echo **graphics** output using **grid** graphics*

Description

Convert a scene that was drawn using the **graphics** package to an identical scene drawn with the **grid** package.

Usage

```
grid.echo(x = NULL, newpage = TRUE, prefix = NULL)
```

Arguments

<code>x</code>	Either NULL, a recorded plot, or a function. If NULL, the current graphics device is echoed.
<code>newpage</code>	A logical value indicating whether to start a new page. If FALSE, echoing occurs in the current grid viewport.
<code>prefix</code>	A character value used as a prefix for all grob and viewport names. The default prefix is "graphics".

Details

If the first argument is a function, it must be a function with zero arguments. If the function needs access to non-global data, use a closure. The function should not call functions that create or destroy graphics devices, or change the current graphics device.

Value

This function is called for its side-effect of drawing on the current graphics device.

Author(s)

Paul Murrell

Examples

```
## Not run:
# Echo existing drawing
plot(1)
grid.echo()

# Echo result of call to a plotting function
plotfun <- function() plot(1:10)
grid.echo(plotfun)

# Echo result of a plotting function (anonymous) into current viewport
grid.newpage()
```

```

pushViewport(viewport(x=0, width=.5, just="left"))
grid.rect(gp=gpar(col=NA, fill="grey"))
grid.echo(function() plot(1:10), newpage=FALSE)

## End(Not run)

```

plotdiff	<i>Test for identical output from grid.echo()</i>
----------	---------------------------------------------------

Description

Functions to generate a scene using the **graphics** package, reproduce the scene using `grid.echo()`, test whether the two results are identical, and report on any differences.

Usage

```

plotdiff(expr, label, dev = "pdf",
          antialias = TRUE, density = 100, width = 7, height = 7)
plotdiffInit()
plotdiffResult(warn = FALSE)

```

Arguments

expr	An expression that draws something using the graphics package.
label	A character value that is used to label files generated during testing.
dev	The graphics device used for drawing and echoing. Currently can only be either "pdf" or "png".
antialias	A logical value indicating whether to perform antialiasing when converting from PDF to PNG.
density	A numeric value indicating the resolution (dpi) to use when converting from PDF to PNG.
width, height	Numeric values indicating the size of the device to test on.
warn	A logical value indicating whether non-identical output should produce a warning or an error.

Details

In default usage, `plotdiff()` is used to generate two PDF files, one using the original expression and the other from a call to `grid.echo()`. The PDF files are then converted to PNG files and the PNG files are compared (using `ImageMagick`). If there are any differences, the comparison generates a further PNG file that shows the differences.

All files are currently generated in the current working directory.

Text messages are also generated by `plotdiff()` whenever a difference is found, but those messages are stored up rather than printed immediately. The `plotdiffResult()` function prints out all messages since the last call to `plotdiffInit()`.

Standard usage involves calling `plotdiffInit()`, followed by one or more `plotdiff()` calls, then finally a call to `plotdiffResult()`. Examples of the usage of these functions are provided in the numerous test scripts in the `test-scripts` directory of the package.

Value

All functions are run for their side effects. In the case of `plotdiff()`, the generation of PDF and PNG files and the accumulation of differences about messages. In the case of `plotdiffResult()`, a print out of the accumulated messages, plus possibly either an error or warning.

Author(s)

Paul Murrell

See Also

[grid.echo](#)

Examples

```
## Not run:  
plotdiff(expression(plot(1)), "plot")  
  
## End(Not run)
```

Index

*Topic **dplot**

grid.echo, [2](#)

plotdiff, [3](#)

grid.echo, [2](#), [4](#)

plotdiff, [3](#)

plotdiffInit (plotdiff), [3](#)

plotdiffResult (plotdiff), [3](#)