

Package: gDRstyle (via r-universe)

September 27, 2026

Type Package

Title A package with style requirements for the gDR suite

Version 1.11.6

Date 2026-07-22

Description Package fills a helper package role for whole gDR suite.
It helps to support good development practices by keeping style requirements and style tests for other packages. It also contains build helpers to make all package requirements met.

License Artistic-2.0

LazyLoad yes

Depends R (>= 4.2)

Imports BiocCheck, BiocManager, BiocStyle, checkmate, data.table, desc, git2r, lintr (>= 3.2.0), rcmdcheck, remotes, yaml, rjson, pkgbuild, withr

Suggests knitr, pkgdown, testthat (>= 3.0.0)

URL <https://github.com/gdrplatform/gDRstyle>,
<https://gdrplatform.github.io/gDRstyle/>

BugReports <https://github.com/gdrplatform/gDRstyle/issues>

biocViews Software, Infrastructure

VignetteBuilder knitr

ByteCompile TRUE

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Encoding UTF-8

Config/pak/sysreqs
cmake git make libgit2-dev libicu-dev libssh2-1-dev libuv1-dev libxml2-dev libssl-dev

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

Date/Publication 2026-07-29 09:06:39 UTC

RemoteUrl <https://github.com/bioc/gDRstyle>

RemoteRef HEAD
RemoteSha 984a92bb85c1bb08dc2c32c189b0f3bdd771fd70

Contents

avoid_new_lines	2
checkDependencies	3
checkPackage	4
installAllDeps	5
installLocalPackage	6
lintNewsEntries	6
lintPkgDirs	7
lintRmdDeps	8
roxygen_tag_linter	9

Index	10
--------------	-----------

avoid_new_lines	<i>Avoid new lines in sprintf output. Function helps to avoid line lenght limits without affecting sprintf output</i>
-----------------	---

Description

Avoid new lines in sprintf output. Function helps to avoid line lenght limits without affecting sprintf output

Usage

avoid_new_lines(fmt)

Arguments

fmt	string, formatted as sprintf input
-----	------------------------------------

Value

string

Examples

```
sprintf(avoid_new_lines(  
  "Lorem ipsum dolor sit amet, %s adipiscing elit, sed do eiusmod  
  tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim  
  veniam,"  
), "consectetur")
```

checkDependencies	<i>Check for alignment of package dependencies across Rplatform and package specifications.</i>
-------------------	---

Description

Check the package dependency version specifications in the rplatform/dependencies.yaml and DESCRIPTION.

Usage

```
checkDependencies(  
  dep_path,  
  desc_path,  
  skip_pkgs = "R",  
  combo_path = "/mnt/vol/dependencies_combo.yaml"  
)
```

Arguments

dep_path	String of path to the rplatform dependencies.yaml file.
desc_path	String of the path to the package DESCRIPTION file.
skip_pkgs	vector of packages from DESCRIPTION to skip; defaults to R
combo_path	String of path to the combo image dependencies.yaml file. Defaults to the current directory.

Details

This function is used for its side effects in the event that there are dependency clashes.

Value

NULL invisibly.

Examples

```
checkDependencies(  
  dep_path =  
    system.file(package = "gDRstyle", "testdata", "dependencies.yaml"),  
  desc_path = system.file(package = "gDRstyle", "DESCRIPTION"),  
  skip_pkgs = c("testthat", "lintr")  
)
```

checkPackage	<i>Check R package</i>
--------------	------------------------

Description

Used in gDR platform packages' CI/CD pipelines to check that the package abides by gDRstyle stylistic requirements, passes rcmdcheck, and ensures that the dependencies.yml file used to build gDR platform's docker image is kept up-to-date with the dependencies listed in the package's DESCRIPTION file.

Usage

```
checkPackage(
  pkgName,
  repoDir,
  subdir = NULL,
  fail_on = "warning",
  bioc_check = FALSE,
  run_examples = TRUE,
  skip_lint = FALSE,
  skip_tests = FALSE,
  skip_pkgdown = FALSE,
  build_vignettes = TRUE,
  check_vignettes = TRUE,
  as_cran = FALSE
)
```

Arguments

pkgName	String of package name.
repoDir	String of path to repository directory.
subdir	String of relative path to the R package root directory from the repoDir.
fail_on	String specifying the level at which check fail. Supported values: "note", "warning" (default) and "error".
bioc_check	Logical whether bioc check should be performed
run_examples	Logical whether examples check should be performed
skip_lint	skip lint checks
skip_tests	skip tests
skip_pkgdown	skip pkgdown build
build_vignettes	build vignettes
check_vignettes	check vignettes
as_cran	run with as_cran flag

Value

NULL invisibly.

Examples

```
checkPackage(
  pkgName = "fakePkg",
  repoDir = system.file(package = "gDRstyle", "tst_pkgs", "dummy_pkg"),
  fail_on = "error"
)
```

installAllDeps	<i>Install all package dependencies from yaml file for building image purposes</i>
----------------	--

Description

Install all package dependencies from yaml file for building image purposes

Usage

```
installAllDeps(
  additionalRepos = NULL,
  base_dir = "/mnt/vol",
  use_ssh = FALSE,
  test_mode = FALSE
)
```

Arguments

additionalRepos	List of additional Repos
base_dir	String of base working directory.
use_ssh	logical, if use ssh keys
test_mode	logical, whether to run the function in the test mode (if TRUE the dependencies are not installed but only listed)

Value

NULL

Examples

```
installAllDeps(
  base_dir = system.file(package = "gDRstyle", "testdata"),
  test_mode = TRUE
)
```

installLocalPackage	<i>Install locally cloned repo for building image purposes</i>
---------------------	--

Description

Install locally cloned repo for building image purposes

Usage

```
installLocalPackage(repo_path, additionalRepos = NULL, base_dir = "/mnt/vol")
```

Arguments

repo_path	String of repository directory.
additionalRepos	List of additional Repos
base_dir	String of base working directory.

Value

NULL

Examples

```
installLocalPackage(system.file(
  package = "gDRstyle", "tst_pkgs", "dummy_pkg"
))
```

lintNewsEntries	<i>Lint NEWS.md entries for style and brevity</i>
-----------------	---

Description

Checks that every bullet entry in NEWS.md follows the gDR style guidelines: starts with an imperative verb, is concise (no verbose phrases, no trailing period, within the character limit), that each version section contains at most max_bullets entries, and that version headers match the expected format.

Usage

```
lintNewsEntries(pkg_dir = ".", max_chars = 120L, max_bullets = 3L)
```

Arguments

pkg_dir	character(1) path to the package root directory containing NEWS.md. Defaults to the current directory.
max_chars	integer(1) maximum number of characters allowed per bullet entry (excluding the leading "* "). Defaults to 120L.
max_bullets	integer(1) maximum number of bullet entries allowed per version section. Defaults to 3L.

Value

NULL invisibly if no violations are found. Stops with an error listing all violations otherwise.

Examples

```
pkg_dir <- system.file(package = "gDRstyle", "tst_pkgs", "dummy_pkg")
lintNewsEntries(pkg_dir)
```

lintPkgDirs	<i>Lint select subdirectories in a package directory.</i>
-------------	---

Description

Lint select subdirectories in a package directory.

Usage

```
lintPkgDirs(pkg_dir = ".", shiny = FALSE)
```

Arguments

pkg_dir	String of path to package directory.
shiny	Boolean of whether or not a shiny directory should also be lint. Defaults to the current directory.

Details

Will look for files in the following directories: "R", "tests", and conditionally "inst/shiny" if shiny is TRUE.

Value

NULL invisibly.

Examples

```
lintPkgDirs(
  pkg_dir= system.file(package = "gDRstyle", "tst_pkgs", "dummy_pkg"))
```

`lintRmdDeps`*Check an Rmd file for unresolved function calls*

Description

Parses all R chunks in an Rmd file, collects packages loaded via `library()` / `require()`, detects the host package from the nearest DESCRIPTION file, identifies locally defined names, and flags any unqualified function call that cannot be resolved to a known symbol.

Usage

```
lintRmdDeps(rmd_path, verbose = FALSE)
```

Arguments

<code>rmd_path</code>	string path to an .Rmd file.
<code>verbose</code>	logical(1) if TRUE, prints diagnostics (host package, loaded packages, local definitions, known symbol count). Default FALSE.

Details

Mustache template syntax (`{{param}}`) is stripped before parsing so parametrised report templates can be checked without syntax errors.

Value

A `data.table` with columns `func` (character) and `line` (integer) - one row per unique unresolved call site. Returns `NULL` when the file contains no R chunks.

Examples

```
## Not run:
# Check a single template
lintRmdDeps("inst/report_templates/3-analysis.Rmd")

# Check with verbose diagnostics
lintRmdDeps("inst/report_templates/3-analysis.Rmd", verbose = TRUE)

## End(Not run)
```

roxygen_tag_linter	<i>roxygen_tag_linter</i>
--------------------	---------------------------

Description

Check that function has documented specific tag in Roxygen skeleton (default @author).

Usage

```
roxygen_tag_linter(tag = "@author")
```

Arguments

tag	character (default @author)
-----	-----------------------------

Value

linter class function

Author(s)

Kamil Foltynski kamil.foltynski@contractors.roche.com

Examples

```
linters_config <- lintr::linters_with_defaults(  
  line_length_linter = lintr::line_length_linter(120),  
  roxygen_tag_linter = roxygen_tag_linter()  
)
```

Index

* **check**

checkDependencies, [3](#)

checkPackage, [4](#)

* **install**

installAllDeps, [5](#)

installLocalPackage, [6](#)

* **linter**

avoid_new_lines, [2](#)

lintNewsEntries, [6](#)

lintPkgDirs, [7](#)

lintRmdDeps, [8](#)

roxygen_tag_linter, [9](#)

avoid_new_lines, [2](#)

checkDependencies, [3](#)

checkPackage, [4](#)

installAllDeps, [5](#)

installLocalPackage, [6](#)

lintNewsEntries, [6](#)

lintPkgDirs, [7](#)

lintRmdDeps, [8](#)

roxygen_tag_linter, [9](#)