

Package ‘cpp11armadillo’

March 21, 2025

Type Package

Title An 'Armadillo' Interface

Description Provides function declarations and inline function definitions that facilitate communication between R and the 'Armadillo' 'C++' library for linear algebra and scientific computing. This implementation is detailed in Vargas Sepulveda and Schneider Malamud (2024) <[doi:10.1016/j.softx.2025.102087](https://doi.org/10.1016/j.softx.2025.102087)>.

Version 0.4.6

Imports cpp11

Suggests desc, knitr, mockery, rmarkdown, testthat (>= 3.0.0), withr

Depends R(>= 3.5.0)

License Apache License (>= 2)

BugReports <https://github.com/pachadotdev/cpp11armadillo/issues>

URL <https://pacha.dev/cpp11armadillo/>,
<https://github.com/pachadotdev/cpp11armadillo>

RoxygenNote 7.3.2

Encoding UTF-8

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Mauricio Vargas Sepulveda [aut, cre]
(<<https://orcid.org/0000-0003-1017-7574>>),
Jonathan Schneider Malamud [ctb],
Conrad Sanderson [aut] (Armadillo library (C++))

Maintainer Mauricio Vargas Sepulveda <m.sepulveda@mail.utoronto.ca>

Repository CRAN

Date/Publication 2025-03-21 06:10:06 UTC

Contents

armadillo_version	2
cpp_vendor	2
pkg_template	3

Index	4
--------------	----------

armadillo_version	<i>Get Armadillo version</i>
-------------------	------------------------------

Description

Provides the Armadillo C++ library version name and number included in the package.

Usage

```
armadillo_version()
```

Value

A string with the Armadillo version name and number

Examples

```
armadillo_version()
```

cpp_vendor	<i>Vendor the cpp11 and cpp11armadillo dependency</i>
------------	---

Description

Vendoring is the act of making your own copy of the 3rd party packages your project is using. It is often used in the go language community.

Usage

```
cpp_vendor(dir = NULL, subdir = "/inst/include")
```

Arguments

dir	The directory to vendor the code into.
subdir	The subdirectory to vendor the code into.

Details

This function vendors `cpp11` and `cpp11armadillo` into your package by copying the `cpp11` and `cpp11armadillo` headers into the `'inst/include'` folder and adding `'cpp11 version: XYZ'` and `'cpp11armadillo version: XYZ'` to the top of the files, where `XYZ` is the version of `cpp11` and `cpp11armadillo` currently installed on your machine.

Vendoring places the responsibility of updating the code on you. Bugfixes and new features in `cpp11` and `cpp11armadillo` will not be available for your code until you run `'cpp_vendor()'` again.

Value

The file path to the vendored code (invisibly).

Examples

```
# create a new directory
dir <- tempdir()
dir.create(dir)

# vendor the cpp11 headers into the directory
cpp_vendor(dir)
```

pkg_template

Start a new project with the cpp11armadillo package template

Description

Start a new project with the `cpp11armadillo` package template

Usage

```
pkg_template(path = NULL, pkgname = NULL)
```

Arguments

<code>path</code>	Path to the new project
<code>pkgname</code>	Name of the new package

Value

The file path to the copied template (invisibly).

Examples

```
# create a new directory
dir <- tempdir()
dir.create(dir)

# copy the package template into the directory
pkg_template(dir, "mynewpkg")
```

Index

`armadillo_version`, [2](#)

`cpp_vendor`, [2](#)

`pkg_template`, [3](#)