

Package: `cpp1eigen` (via `r-universe`)

September 11, 2024

Type Package

Title An 'Eigen' Interface

Description Provides function declarations and inline function definitions that facilitate communication between R and the 'Eigen' 'C++' library for linear algebra and scientific computing.

Version 0.2.5

Suggests `cpp11`, `desc`, `knitr`, `mockery`, `rmarkdown`, `testthat` ($\geq 3.0.0$),
withr

Depends `R` ($\geq 3.5.0$)

License Apache License (≥ 2)

BugReports <https://github.com/pachadotdev/cpp1eigen/issues>

URL <https://pacha.dev/cpp1eigen/>,
<https://github.com/pachadotdev/cpp1eigen>

LazyData true

RoxygenNote 7.3.1

Encoding UTF-8

VignetteBuilder knitr

Config/testthat/edition 3

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

RemoteUrl <https://github.com/pachadotdev/cpp1eigen>

RemoteRef HEAD

RemoteSha 739ccd2dc826328a9f246bf4e770759dc1931a4d

Contents

<code>cpp_vendor</code>	2
<code>eigen_version</code>	3
<code>pkg_template</code>	3

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

`cpp_vendor`*Vendor the cpp11 and cpp11eigen dependency*

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

<code>dir</code>	The directory to vendor the code into.
<code>subdir</code>	The subdirectory to vendor the code into.

Details

This function vendors `cpp11` and `cpp11eigen` into your package by copying the `cpp11` and `cpp11eigen` headers into the `'inst/include'` folder and adding `'cpp11 version: XYZ'` and `'cpp11eigen version: XYZ'` to the top of the files, where `XYZ` is the version of `cpp11` and `cpp11eigen` currently installed on your machine.

Vendoring places the responsibility of updating the code on you. Bugfixes and new features in `cpp11` and `cpp11eigen` 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)
```

eigen_version	<i>Get eigen version</i>
---------------	--------------------------

Description

Provides the eigen C++ library version number included in the package.

Usage

```
eigen_version()
```

Value

A string with the eigen version name and number

Examples

```
eigen_version()
```

pkg_template	<i>Start a new project with the cpp11 eigen package template</i>
--------------	--

Description

Start a new project with the cpp11 eigen package template

Usage

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

Arguments

path	Path to the new project
pkgname	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

`cpp_vendor`, 2

`eigen_version`, 3

`pkg_template`, 3