

# Package ‘cpp11eigen’

May 8, 2026

**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** 0.2.0

**Suggests** cpp11, desc, knitr, mockery, rmarkdown, testthat (>= 3.0.0),  
withr

**Depends** R(>= 3.5.0)

**License** Apache License (>= 2)

**BugReports** <https://github.com/pachadotdev/cpp11eigen/issues>

**URL** <https://pacha.dev/cpp11eigen/>,  
<https://github.com/pachadotdev/cpp11eigen>

**RoxygenNote** 7.3.1

**Encoding** UTF-8

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Mauricio Vargas Sepulveda [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-1017-7574>>),  
Gael Guennebaud [aut] (Eigen library (C++)),  
Gael Guennebaud [aut] (Eigen library (C++)),  
Benot Jacob [aut] (Eigen library (C++)),  
Intel Corporation [aut] (Eigen library (C++)),  
Xerox Corporation [aut] (Eigen library (C++))

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

**Repository** CRAN

**Date/Publication** 2024-09-04 13:40:08 UTC

## Contents

cpp_vendor	2
eigen_version	3
pkg_template	3
<b>Index</b>	<b>4</b>

---

cpp_vendor	<i>Vendor the cpp11 and cpp11eigen 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 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