

# Package: RcppInt64 (via r-universe)

August 28, 2024

**Type** Package

**Title** 'Rcpp'-Based Helper Functions to Pass 'Int64' and 'nanotime'  
Values Between 'R' and 'C++'

**Version** 0.0.5

**Date** 2024-04-30

**Description** 'Int64' values can be created and accessed via the 'bit64' package and its 'integer64' class which package the 'int64' representation cleverly into a 'double'. The 'nanotime' packages builds on this to support nanosecond-resolution timestamps. This packages helps conversions between 'R' and 'C++' via several helper functions provided via a single header file. A complete example client package is included as an illustration.

**URL** <https://github.com/eddelbuettel/rcppint64>

**BugReports** <https://github.com/eddelbuettel/rcppint64/issues>

**License** GPL (>= 2)

**Imports** Rcpp (>= 1.0.8)

**LinkingTo** Rcpp

**Suggests** tinytest, bit64, nanotime

**RoxygenNote** 6.0.1

**Encoding** UTF-8

**Repository** <https://eddelbuettel.r-universe.dev>

**RemoteUrl** <https://github.com/eddelbuettel/rcppint64>

**RemoteRef** HEAD

**RemoteSha** 4cf8fe1d511cf9f18affaab597d9f8a2d3807853

## Contents

RcppInt64-package . . . . .	2
Int64toInt64 . . . . .	2
NanotimeToNanotime . . . . .	3

**Index****4**


---

RcppInt64-package	<i>'Rcpp'-Based Helper Functions to Pass 'Int64' and 'nanotime' Values Between 'R' and 'C++'</i>
-------------------	--

---

**Description**

'Int64' values can be created and accessed via the 'bit64' package and its 'integer64' class which package the 'int64' representation cleverly into a 'double'. The 'nanotime' packages builds on this to support nanosecond-resolution timestamps. This packages helps conversions between 'R' and 'C++' via several helper functions provided via a single header file. A complete example client package is included as an illustration.

**Package Content**

Index: This package was not yet installed at build time.

**Maintainer**

Dirk Eddelbuettel <edd@debian.org>

**Author(s)**

Dirk Eddelbuettel [aut, cre] (<<https://orcid.org/0000-0001-6419-907X>>)

---

Int64toInt64	<i>Integer64 to Integer64 round-trip demo</i>
--------------	---

---

**Description**

This function takes an integer64-valued input vector, converts it to the equivalent int64\_t vector in C++, displays each element after first adding one, and returns the modified vector.

**Usage**

```
Int64toInt64(vec)
```

**Arguments**

vec	An integer64-classed vector from R
-----	------------------------------------

**Value**

A modified integer64 vector where each element increased by one

**Examples**

```
# generate all powers of 10 fro 0 .. 18
if (requireNamespace("bit64", quietly=TRUE)) {
  v <- bit64::as.integer64(10^seq(0,18))
  # pass them to function which will add one to each, print and return
  Int64toInt64(v)
}
```

---

NanotimeToNanotime	<i>nanotime to nanotime round-trip demo</i>
--------------------	---

---

**Description**

This function takes an nanotime-valued input vector, converts it to the equivalent int64\_t vector in C++, displays each element after first adding one, and returns the modified vector.

**Usage**

```
NanotimeToNanotime(vec)
```

**Arguments**

vec	A nanotime-classed vector from R
-----	----------------------------------

**Value**

A modified nanotime vector where each element increased by one

**Examples**

```
# generate all powers of 10 fro 0 .. 18
if (requireNamespace("nanotime", quietly=TRUE)) {
  v <- nanotime::as.nanotime(10^seq(0,18))
  # pass them to function which will add one to each, print and return
  NanotimeToNanotime(v)
}
```

# Index

\* **package**

RcppInt64-package, [2](#)

Int64toInt64, [2](#)

NanotimeToNanotime, [3](#)

RcppInt64 (RcppInt64-package), [2](#)

RcppInt64-package, [2](#)