

# Package: serializer (via r-universe)

November 1, 2024

**Type** Package

**Title** Expose R's Serialization Interface

**Version** 0.2.2

**Maintainer** Mike Cheng <mikefc@coolbutuseless.com>

**Description** Expose R's serialization interface for serialing/unserializing R objects to/from raw vectors and connections.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**Suggests** testthat

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

**RemoteUrl** <https://github.com/coolbutuseless/serializer>

**RemoteRef** HEAD

**RemoteSha** dd15f4373d62c0bf8530b411e0db3e656213c7c1

## Contents

calc_serialized_size . . . . .	2
marshall_con . . . . .	2
marshall_con_illegal . . . . .	3
marshall_file . . . . .	3
marshall_raw . . . . .	4
unmarshall_con . . . . .	5
unmarshall_con_illegal . . . . .	5
unmarshall_file . . . . .	6
unmarshall_raw . . . . .	6

<b>Index</b>	<b>8</b>
--------------	----------

calc\_serialized\_size *Calculate the size of the R object when it is serialized*

---

**Description**

This function goes through all the motions of serializing an object, but does nothing with the bytes other than to tally the total length.

**Usage**

```
calc_serialized_size(robj)
```

**Arguments**

robj            R object

**Value**

number of bytes needed to serialize this object

**Examples**

```
calc_serialized_size(mtcars)
```

---

marshall\_con            *Serialize an R object to a connection*

---

**Description**

Serialize an R object to a connection

**Usage**

```
marshall_con(robj, con = NULL)
```

**Arguments**

robj            R object  
con            connection to write the data to

**Value**

none

**Examples**

```
tmp <- tempfile()
marshall_con(mtcars, con = gzfile(tmp))
unmarshall_con(gzfile(tmp))
```

---

marshall\_con\_illegal    *Serialize an R object to a connection using a non-API function from R*

---

**Description**

This approach will not pass R CMD check as the function 'R\_WriteConnection()' is not part of the R API, but it is much much faster than the officially blessed method of calling-back-into-R to write to a connection.

**Usage**

```
marshall_con_illegal(rojb, con = NULL)
```

**Arguments**

rojb	R object
con	connection to write the data to

**Value**

none

**Examples**

```
tmp <- tempfile()
marshall_con_illegal(mtcars, con = gzfile(tmp))
unmarshall_con_illegal(gzfile(tmp))
```

---

marshall\_file            *Serialize an R object to a file*

---

**Description**

Serialize an R object to a file

**Usage**

```
marshall_file(rojb, filename)
```

**Arguments**

robj	R object
filename	filename

**Value**

None

**Examples**

```
tmp <- tempfile()
marshall_file(head(mtcars), tmp)
unmarshall_file(tmp)
```

---

marshall_raw	<i>Serialize an R object to a raw vector</i>
--------------	--

---

**Description**

Serialize an R object to a raw vector

**Usage**

```
marshall_raw(robj)
```

**Arguments**

robj	R object
------	----------

**Value**

a raw vector containing the serialized data.

**Examples**

```
m <- marshall_raw(head(mtcars))
m
unmarshall_raw(m)
```

---

unmarshall_con	<i>Unserialize an R object from a connection</i>
----------------	--

---

**Description**

Unserialize an R object from a connection

**Usage**

```
unmarshall_con(con)
```

**Arguments**

con                    A connection from which to read serialized data

**Value**

Return the unserialized R object

**Examples**

```
tmp <- tempfile()
marshall_con(mtcars, con = gzfile(tmp))
unmarshall_con(gzfile(tmp))
```

---

unmarshall_con_illegal
------------------------

*Unserialize an R object from a connection using a non-API function from R*

---

**Description**

This approach will not pass R CMD check as the function 'R\_ReadConnection()' is not part of the R API, but it is much much faster than the officially blessed method of calling-back-into-R to read from a connection.

**Usage**

```
unmarshall_con_illegal(con)
```

**Arguments**

con                    A connection from which to read serialized data

**Value**

Return the unserialized R object

**Examples**

```
tmp <- tempfile()
marshall_con_illegal(mtcars, con = gzfile(tmp))
unmarshall_con_illegal(gzfile(tmp))
```

---

unmarshall_file	<i>Unserialize an R object from a file</i>
-----------------	--

---

**Description**

Unserialize an R object from a file

**Usage**

```
unmarshall_file(filename)
```

**Arguments**

filename            a file containing serialized data

**Value**

Return the unserialized R object

**Examples**

```
tmp <- tempfile()
marshall_file(head(mtcars), tmp)
unmarshall_file(tmp)
```

---

unmarshall_raw	<i>Unserialize an R object from a raw vector</i>
----------------	--

---

**Description**

Unserialize an R object from a raw vector

**Usage**

```
unmarshall_raw(raw_vec)
```

**Arguments**

raw\_vec            A raw vector containing serialized data

**Value**

Return the unserialized R object

**Examples**

```
m <- marshall_raw(head(mtcars))
m
unmarshall_raw(m)
```

# Index

`calc_serialized_size`, 2

`marshall_con`, 2

`marshall_con_illegal`, 3

`marshall_file`, 3

`marshall_raw`, 4

`unmarshall_con`, 5

`unmarshall_con_illegal`, 5

`unmarshall_file`, 6

`unmarshall_raw`, 6