

# Package ‘mori’

May 9, 2026

**Title** Shared Memory for R Objects

**Version** 0.2.0

**Description** Share R objects across processes on the same machine via a single copy in 'POSIX' shared memory (Linux, macOS) or a 'Win32' file mapping (Windows). Every process reads from the same physical pages through the R Alternative Representation ('ALTREP') framework, giving lazy, zero-copy access. Shared objects serialize compactly as their shared memory name rather than their full contents.

**License** MIT + file LICENSE

**URL** <https://shikokuchuo.net/mori/>, <https://github.com/shikokuchuo/mori>

**BugReports** <https://github.com/shikokuchuo/mori/issues>

**Depends** R (>= 4.3)

**Suggests** lobstr, mirai, testthat (>= 3.0.0)

**Config/build/compilation-database** true

**Config/roxygen2/markdown** TRUE

**Config/roxygen2/version** 8.0.0

**Config/testthat/edition** 3

**Encoding** UTF-8

**NeedsCompilation** yes

**Author** Charlie Gao [aut, cre] (ORCID: <<https://orcid.org/0000-0002-0750-061X>>),  
Posit Software, PBC [cph, fnd] (ROR: <<https://ror.org/03wc8by49>>)

**Maintainer** Charlie Gao <[charlie.gao@posit.co](mailto:charlie.gao@posit.co)>

**Repository** CRAN

**Date/Publication** 2026-05-09 01:41:57 UTC

## Contents

mori-package . . . . .	2
is_shared . . . . .	2

map_shared . . . . .	3
share . . . . .	4
shared_name . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

mori-package	<i>mori: Shared Memory for R Objects</i>
--------------	--

---

## Description

Share R objects via shared memory with `share()`, access them in other processes with `map_shared()`, using R's ALTREP framework for zero-copy memory-mapped access. Shared objects serialize compactly via ALTREP serialization hooks. Shared memory is automatically freed when the R object is garbage collected.

## Author(s)

**Maintainer:** Charlie Gao <charlie.gao@posit.co> ([ORCID](#))

Authors:

- Charlie Gao <charlie.gao@posit.co> ([ORCID](#))

Other contributors:

- Posit Software, PBC ([ROR](#)) [copyright holder, funder]

## See Also

Useful links:

- <https://shikokuchuo.net/mori/>
- <https://github.com/shikokuchuo/mori>
- Report bugs at <https://github.com/shikokuchuo/mori/issues>

---

is_shared	<i>Test if an Object is Shared</i>
-----------	------------------------------------

---

## Description

Returns TRUE if x is an ALTREP object backed by shared memory (created by `share()` or `map_shared()`), FALSE otherwise.

## Usage

```
is_shared(x)
```

**Arguments**

x                    an R object.

**Value**

TRUE or FALSE.

**Examples**

```
x <- share(rnorm(100))
is_shared(x)
is_shared(rnorm(100))
```

---

map\_shared

*Open Shared Memory by Name*

---

**Description**

Open a shared memory region identified by a name string and return an ALTREP-backed R object that reads directly from shared memory.

**Usage**

```
map_shared(name)
```

**Arguments**

name                a character string as returned by [shared\\_name\(\)](#): either a bare shared memory name (opens the root) or a name with a 1-based bracketed index path (e.g. `"/mori_abc_1[2,3]"`, opens the addressed sub-list or element directly).

**Value**

The R object stored at the named region (or sub-object at the given path), or NULL if name is not a valid shared memory name (wrong type, length, NA, missing or malformed prefix, or malformed bracketed path). If name parses as valid but the region is absent or corrupted — or the path doesn't address a valid sub-object — an error is raised.

**See Also**

[share\(\)](#) to create a shared object, [shared\\_name\(\)](#) to extract the shared memory name.

**Examples**

```
x <- share(1:100)
nm <- shared_name(x)
y <- map_shared(nm)
sum(y)
```

---

share

*Create a Shared Object*

---

## Description

Write an R object into shared memory and return a version that other processes on the same machine can map without copying.

## Usage

```
share(x)
```

## Arguments

x                    an R object.

## Details

Attributes are stored alongside the data in the shared memory region and restored on the consumer side. Character vectors use a packed layout and elements are materialised lazily on access. When serialised (e.g. by `serialize()` or across a `mirai()` call), a shared object is represented compactly by its shared memory name (~30 bytes) rather than by its contents.

The shared memory region is managed automatically. It stays alive as long as the returned object (or any element extracted from it) is referenced in R, and is freed by the garbage collector when no references remain.

`share()` is idempotent: calling it on an object that is already backed by shared memory returns the input unchanged without allocating a new region.

**Important:** always assign the result of `share()` to a variable. The shared memory is kept alive by the R object reference — if the result is used as a temporary (not assigned), the garbage collector may free the shared memory before a consumer process has mapped it.

## Value

For atomic vectors (including character vectors and those with attributes such as `names`, `dim`, `class`, or `levels`) and lists or data frames whose elements are such vectors, an ALTREP-backed object that reads directly from shared memory. For any other object (environments, closures, language objects, `NULL`), the input is returned unchanged with no shared memory region created.

## Persistence

Direct `saveRDS()` of a shared object writes only the shared memory name, so the resulting file is meaningful only on the same machine while the region is still alive. For portable storage or transport across machines, materialise into a regular in-memory copy first with `rlang::duplicate()`, which deep-duplicates the object:

```
saveRDS(rlang::duplicate(x), file = "x.rds")
```

**See Also**

[map\\_shared\(\)](#) to open a shared region by name, [shared\\_name\(\)](#) to extract the shared memory name.

**Examples**

```
x <- share(rnorm(100))
sum(x)
```

---

shared_name	<i>Extract Shared Memory Name</i>
-------------	-----------------------------------

---

**Description**

Extract the shared memory name from a shared object. This name can be passed to [map\\_shared\(\)](#) to open the same region in another process.

**Usage**

```
shared_name(x)
```

**Arguments**

x a shared object as returned by [share\(\)](#) or [map\\_shared\(\)](#).

**Value**

A character string identifying the shared object, or NULL if x is not a shared object. For a sub-list or element extracted from a shared list, the string carries a bracketed 1-based index path (e.g. `"/mori_abc_1[2,3]"`). [map\\_shared\(\)](#) accepts both forms; the path-qualified form returns the addressed sub-object directly. The underlying shared memory region name is the prefix before `[` and is recoverable via `sub("\\[.*$", "", shared_name(x))`.

**See Also**

[map\\_shared\(\)](#) to open a shared region by name.

**Examples**

```
x <- share(rnorm(100))
shared_name(x)
```

# Index

`is_shared`, 2

`map_shared`, 3

`map_shared()`, 2, 5

`mori` (`mori-package`), 2

`mori-package`, 2

`saveRDS()`, 4

`serialize()`, 4

`share`, 4

`share()`, 2, 3, 5

`shared_name`, 5

`shared_name()`, 3, 5