

Package ‘mangoro’

November 25, 2025

Title 'R'/Go 'IPC' using 'Nanomsg' Next Gen

Version 0.2.6

Copyright See inst/AUTHORS

Description The tools provide an interface for inter-process communication ('IPC') between 'R' and 'Go' using the 'Nanomsg' library. It vendors on the 'Go' side the 'mangos' library <<https://github.com/nanomsg/mangos>> and 'arrow-go' <<https://github.com/apache/arrow-go>> and uses 'nanonext' and 'nanoarrow' on the 'R' side.

License GPL-3

Depends R (>= 4.4.0)

Imports nanonext, nanoarrow, jsonlite

Suggests processx

SystemRequirements Go (<https://golang.org>), GNU make

URL <https://github.com/souunkou-bioinfo/mangoro>,
<https://souunkou-bioinfo.github.io/mangoro/>

BugReports <https://github.com/souunkou-bioinfo/mangoro/issues>

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Souunkou Mahamane Toure [aut, cre],
Mangos (nanomsg) contributors [cph],
The Go Authors [cph],
Apache Arrow Project [cph]

Maintainer Souunkou Mahamane Toure <souunkoutoure@gmail.com>

Repository CRAN

Date/Publication 2025-11-25 20:52:10 UTC

Contents

create_ipc_path	2
find_go	3
find_mangoro_vendor	3
get_arrow_go_version	3
get_mangos_version	4
mangoro_go_build	4
mangoro_http_start	5
mangoro_http_status	6
mangoro_http_stop	6
mangoro_pack_int32	7
mangoro_rpc_call	7
mangoro_rpc_call_message	8
mangoro_rpc_get_manifest	8
mangoro_rpc_manifest_request	9
mangoro_rpc_parse_response	9
mangoro_rpc_recv	10
mangoro_rpc_send	10
mangoro_unpack_int32	11
Index	12

create_ipc_path	Create a unique IPC path for mangoro
-----------------	--------------------------------------

Description

Create a unique IPC path for mangoro

Usage

```
create_ipc_path(prefix = "mangoro-echo")
```

Arguments

prefix	Prefix for the temp file (default: "mangoro-echo")
--------	--

Value

IPC URL string suitable for nanonext and mangoro Go binaries

find_go	<i>Find the path to the Go executable</i>
---------	---

Description

Find the path to the Go executable

Usage

find_go()

Value

Path to the Go binary

find_mangoro_vendor	<i>Find the path to the mangoro vendor directory</i>
---------------------	--

Description

Find the path to the mangoro vendor directory

Usage

find_mangoro_vendor()

Value

Path to the vendor directory (inst/go/vendor)

get_arrow_go_version	<i>Get the version of vendored Arrow Go using Go tooling (no jsonlite)</i>
----------------------	--

Description

Get the version of vendored Arrow Go using Go tooling (no jsonlite)

Usage

get_arrow_go_version()

Value

The version string of github.com/apache/arrow/go/v18 in the vendor go.mod

get_mangos_version	<i>Get the version of vendored mangos using Go tooling (no jsonlite)</i>
--------------------	--

Description

Get the version of vendored mangos using Go tooling (no jsonlite)

Usage

```
get_mangos_version()
```

Value

The version string of go.nanomsg.org/mangos/v3 in the vendor go.mod

mangoro_go_build	<i>Compile a Go source file using the vendored dependencies</i>
------------------	---

Description

Compile a Go source file using the vendored dependencies

Usage

```
mangoro_go_build(src, out, gomaxprocs = 1, gocache = NULL, ...)
```

Arguments

src	Path to the Go source file
out	Path to the output binary
gomaxprocs	Number of threads for Go build (sets GOMAXPROCS env variable)
gocache	Path to Go build cache directory. If NULL (default), uses a temporary directory to comply with CRAN policy. Set to NA to use the default Go cache location.
...	Additional arguments to pass to Go build

Value

Path to the compiled binary

mangoro_http_start	<i>Start an HTTP file server via RPC</i>
--------------------	--

Description

Start an HTTP file server via RPC

Usage

```
mangoro_http_start(  
    sock,  
    addr,  
    dir = ".",  
    prefix = "/",  
    cors = FALSE,  
    coop = FALSE,  
    tls = FALSE,  
    cert = NULL,  
    key = NULL,  
    silent = FALSE  
)
```

Arguments

sock	A nanonext socket connected to the HTTP server controller
addr	Address to bind server to (e.g., "127.0.0.1:8080")
dir	Directory to serve (default: current directory)
prefix	URL prefix for the server (default: "/")
cors	Enable CORS headers (default: FALSE)
coop	Enable Cross-Origin-Opener-Policy (default: FALSE)
tls	Enable TLS (default: FALSE)
cert	Path to TLS certificate file (required if tls = TRUE)
key	Path to TLS key file (required if tls = TRUE)
silent	Suppress server logs (default: FALSE)

Value

List with status and message

mangoro_http_status	<i>Get HTTP server status via RPC</i>
---------------------	---------------------------------------

Description

Get HTTP server status via RPC

Usage

mangoro_http_status(sock)

Arguments

sock	A nanonext socket connected to the HTTP server controller
------	---

Value

List with status and message

mangoro_http_stop	<i>Stop the HTTP file server via RPC</i>
-------------------	--

Description

Stop the HTTP file server via RPC

Usage

mangoro_http_stop(sock)

Arguments

sock	A nanonext socket connected to the HTTP server controller
------	---

Value

List with status and message

mangoro_pack_int32	<i>Pack a 32-bit integer to raw bytes (big-endian)</i>
--------------------	--

Description

Pack a 32-bit integer to raw bytes (big-endian)

Usage

```
mangoro_pack_int32(x)
```

Arguments

x	An integer value
---	------------------

Value

A raw vector of length 4

mangoro_rpc_call	<i>Call a remote function via RPC</i>
------------------	---------------------------------------

Description

Call a remote function via RPC

Usage

```
mangoro_rpc_call(sock, func_name, data)
```

Arguments

sock	A nanonext socket connected to the RPC server
func_name	Name of the function to call
data	Data frame or Arrow stream to send as arguments

Value

The result from `nanoarrow::read_nanoarrow` (typically a `nanoarrow_array_stream`)

`mangoro_rpc_call_message`*Create an RPC function call message*

Description

Create an RPC function call message

Usage

```
mangoro_rpc_call_message(func_name, data)
```

Arguments

<code>func_name</code>	Name of the function to call
<code>data</code>	Data frame or Arrow stream to send as arguments

Value

A raw vector containing the RPC call message

`mangoro_rpc_get_manifest`*Get the manifest of registered functions from an RPC server*

Description

Get the manifest of registered functions from an RPC server

Usage

```
mangoro_rpc_get_manifest(sock)
```

Arguments

<code>sock</code>	A nanonext socket connected to the RPC server
-------------------	---

Value

A list of function signatures

`mangoro_rpc_manifest_request`*Create an RPC manifest request message*

Description

Create an RPC manifest request message

Usage

```
mangoro_rpc_manifest_request()
```

Value

A raw vector containing the manifest request

`mangoro_rpc_parse_response`*Parse an RPC response message*

Description

Parse an RPC response message

Usage

```
mangoro_rpc_parse_response(response)
```

Arguments

response	Raw vector containing the RPC response
----------	--

Value

A list with components: type, func_name, error_msg, data

mangoro_rpc_rcv	<i>Receive a message with retries</i>
-----------------	---------------------------------------

Description

Receive a message with retries

Usage

mangoro_rpc_rcv(sock, max_attempts = 20)

Arguments

- sock A nanonext socket
- max_attempts Maximum number of retry attempts (default 20)

Value

The received message as a raw vector

mangoro_rpc_send	<i>Send a message with retries</i>
------------------	------------------------------------

Description

Send a message with retries

Usage

mangoro_rpc_send(sock, msg, max_attempts = 20)

Arguments

- sock A nanonext socket
- msg Message to send (raw vector)
- max_attempts Maximum number of retry attempts (default 20)

Value

The result from nanonext::send

mangoro_unpack_int32 *Unpack a 32-bit integer from raw bytes (big-endian)*

Description

Unpack a 32-bit integer from raw bytes (big-endian)

Usage

```
mangoro_unpack_int32(bytes)
```

Arguments

bytes	A raw vector of length 4
-------	--------------------------

Value

An integer value

Index

`create_ipc_path`, [2](#)

`find_go`, [3](#)

`find_mangoro_vendor`, [3](#)

`get_arrow_go_version`, [3](#)

`get_mangos_version`, [4](#)

`mangoro_go_build`, [4](#)

`mangoro_http_start`, [5](#)

`mangoro_http_status`, [6](#)

`mangoro_http_stop`, [6](#)

`mangoro_pack_int32`, [7](#)

`mangoro_rpc_call`, [7](#)

`mangoro_rpc_call_message`, [8](#)

`mangoro_rpc_get_manifest`, [8](#)

`mangoro_rpc_manifest_request`, [9](#)

`mangoro_rpc_parse_response`, [9](#)

`mangoro_rpc_recv`, [10](#)

`mangoro_rpc_send`, [10](#)

`mangoro_unpack_int32`, [11](#)