Package: mangoro (via r-universe)

May 14, 2026

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

Version 0.2.15.9000

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, tinytest

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

URL https://github.com/sounkou-bioinfo/mangoro, https://sounkou-bioinfo.github.io/mangoro/

BugReports https://github.com/sounkou-bioinfo/mangoro/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libzstd-dev

Repository https://sounkou-bioinfo.r-universe.dev

Date/Publication 2026-02-13 13:28:38 UTC

RemoteUrl https://github.com/sounkou-bioinfo/mangoro

RemoteRef HEAD

RemoteSha a691d882654dc76645f850b9d269441e2086332c

Contents

`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_mangoro_vendor *Find the path to the mangoro vendor directory*

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

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

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

go_binary_candidates *Determine candidate Go binaries*

Description

Builds a list of candidate go paths from package options, environment variables, PATH entries, and platform-specific defaults. This function does not validate candidates.

Usage

```
go_binary_candidates()
```

Value

Character vector of candidate Go binary paths

mangoro_go_build *Compile a Go source file using the vendored dependencies*

Description

Compiles a Go source file using the vendored dependencies from the mangoro package.

To comply with CRAN policy, this function temporarily redirects several environment variables to prevent Go from writing to user directories:

- `HOME` is set to a temporary directory because Go's telemetry system (introduced in Go 1.23+) writes data to `~/.config/go/telemetry` using `os.UserConfigDir()`, which cannot be disabled via environment variables alone.
- `GOCACHE` is set to a temporary directory to prevent build cache writes to `~/.cache/go-build`.
- `GOENV` is set to a temporary file to prevent config writes to `~/.config/go/env`.

All environment variables are restored and temporary directories cleaned up after the build completes.

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

See Also

https://go.dev/doc/telemetry for Go telemetry documentation

---

mangoro_http_start *Start an HTTP file server via RPC*

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 *Get HTTP server status via RPC*

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 *Stop the HTTP file server via RPC*

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_min_go_version *Find the path to the Go executable*

Description

Locates a usable go binary for runtime IPC helpers. Resolution order:

1. `options(mangoro.go_path)`
2. `Sys.getenv("MANGORO_GO")`
3. `PATH` entries and platform defaults via `go_binary_candidates()`

Candidates are validated by running `go version` and checking the minimum required Go version from the vendored `go.mod`. Errors reference the detected OS/arch using user-friendly labels (e.g., macOS arm64).

Usage

```
mangoro_min_go_version()
```

Value

Path to the Go binary

mangoro_pack_int32 *Pack a 32-bit integer to raw bytes (big-endian)*

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 *Call a remote function via RPC*

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

func_name	Name of the function to call
data	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

`sock`	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_recv *Receive a message with retries*

Description

Receive a message with retries

Usage

```
mangoro_rpc_recv(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 *Send a message with retries*

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