

Package: rho.async (via r-universe)

July 21, 2026

Type Package

Depends R (>= 4.4.0)

Title Rho Async Task and Stream Contracts

Version 0.0.1.9001

Description Defines S7 task, stream, cancellation, and composition contracts for asynchronous R interfaces. Adapters connect nanonext asynchronous handles and promises while keeping every blocking wait explicit.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports S7, s7contract, nanonext, coro, later, promises

Suggests tinytest, knitr

URL <https://github.com/RGenomicsETL/Rho>,
<https://r genomicsetl.github.io/Rho/rho.async/>

BugReports <https://github.com/RGenomicsETL/Rho/issues>

Config/roxygen2/version 8.0.0

Repository <https://r genomicsetl.r-universe.dev>

Date/Publication 2026-07-21 22:15:31 UTC

RemoteUrl <https://github.com/RGenomicsETL/Rho>

RemoteRef HEAD

RemoteSha f3475cf000a379732d37cfb317754dec5ef2b4db

RemoteSubdir packages/rho.async

Contents

rho_async_contracts	2
Index	3

rho_async_contracts	<i>Asynchronous task and stream contracts</i>
---------------------	---

Description

Rho tasks represent one eventual value, while Rho streams represent an asynchronous sequence of typed stream items. Effectful callers compose tasks with `rho_then()` or coroutines and block only at `rho_await()`.

Details

`rho_new_state()` and the concrete adapter task classes are exported for packages implementing new async backends. Polling actions return one of the typed `RhoPollDecision` subclasses; timing remains inside `rho.async`. `rho_serial_queue()` orders asynchronous actions without blocking the event loop. Cancelling a queued task removes only that entry; cancelling active work propagates to the task returned by its action.

Index

`rho_all` (`rho_async_contracts`), [2](#)
`rho_as_promise` (`rho_async_contracts`), [2](#)
`rho_as_task` (`rho_async_contracts`), [2](#)
`rho_async_contracts`, [2](#)
`rho_await` (`rho_async_contracts`), [2](#)
`rho_await`(), [2](#)
`rho_cancel` (`rho_async_contracts`), [2](#)
`rho_catch` (`rho_async_contracts`), [2](#)
`rho_contract_violation`
 (`rho_async_contracts`), [2](#)
`rho_coro_task` (`rho_async_contracts`), [2](#)
`rho_duplex_send` (`rho_async_contracts`), [2](#)
`rho_enqueue` (`rho_async_contracts`), [2](#)
`rho_is_stream` (`rho_async_contracts`), [2](#)
`rho_is_task` (`rho_async_contracts`), [2](#)
`rho_later_event_pump`
 (`rho_async_contracts`), [2](#)
`rho_list_stream` (`rho_async_contracts`), [2](#)
`rho_new_state` (`rho_async_contracts`), [2](#)
`rho_pending` (`rho_async_contracts`), [2](#)
`rho_poll` (`rho_async_contracts`), [2](#)
`rho_poll_complete`
 (`rho_async_contracts`), [2](#)
`rho_poll_failed` (`rho_async_contracts`), [2](#)
`rho_poll_pending` (`rho_async_contracts`),
 [2](#)
`rho_pump_events` (`rho_async_contracts`), [2](#)
`rho_race` (`rho_async_contracts`), [2](#)
`rho_register_task_callback`
 (`rho_async_contracts`), [2](#)
`rho_rejected` (`rho_async_contracts`), [2](#)
`rho_remove_task_callback`
 (`rho_async_contracts`), [2](#)
`rho_serial_queue` (`rho_async_contracts`),
 [2](#)
`rho_signal_contract_violation`
 (`rho_async_contracts`), [2](#)
`rho_stream_close` (`rho_async_contracts`),
 [2](#)
`rho_stream_collect`
 (`rho_async_contracts`), [2](#)
`rho_stream_end` (`rho_async_contracts`), [2](#)
`rho_stream_flat_map`
 (`rho_async_contracts`), [2](#)
`rho_stream_from_task`
 (`rho_async_contracts`), [2](#)
`rho_stream_map` (`rho_async_contracts`), [2](#)
`rho_stream_next` (`rho_async_contracts`), [2](#)
`rho_stream_value` (`rho_async_contracts`),
 [2](#)
`rho_task` (`rho_async_contracts`), [2](#)
`rho_task_callback_bridge`
 (`rho_async_contracts`), [2](#)
`rho_task_from_function`
 (`rho_async_contracts`), [2](#)
`rho_task_from_promise`
 (`rho_async_contracts`), [2](#)
`rho_then` (`rho_async_contracts`), [2](#)
`rho_then`(), [2](#)
`rho_timeout` (`rho_async_contracts`), [2](#)
`rho_wrap_aio` (`rho_async_contracts`), [2](#)
`RhoAsyncError` (`rho_async_contracts`), [2](#)
`RhoAwaitable` (`rho_async_contracts`), [2](#)
`RhoCancellation` (`rho_async_contracts`), [2](#)
`RhoDuplex` (`rho_async_contracts`), [2](#)
`RhoDuplexChannel` (`rho_async_contracts`),
 [2](#)
`RhoEventPump` (`rho_async_contracts`), [2](#)
`RhoEventPumpError`
 (`rho_async_contracts`), [2](#)
`RhoEventPumpIdle` (`rho_async_contracts`),
 [2](#)
`RhoEventPumpProgress`
 (`rho_async_contracts`), [2](#)
`RhoEventPumpResult`
 (`rho_async_contracts`), [2](#)
`RhoEventPumpUnsupported`
 (`rho_async_contracts`), [2](#)

RhoFlatMappedStream
 (rho_async_contracts), [2](#)
RhoFunctionTask (rho_async_contracts), [2](#)
RhoImmediateTask (rho_async_contracts),
 [2](#)
RhoLaterEventPump
 (rho_async_contracts), [2](#)
RhoListStream (rho_async_contracts), [2](#)
RhoMappedStream (rho_async_contracts), [2](#)
RhoNanonextAioTask
 (rho_async_contracts), [2](#)
RhoPollComplete (rho_async_contracts), [2](#)
RhoPollDecision (rho_async_contracts), [2](#)
RhoPollFailed (rho_async_contracts), [2](#)
RhoPollPending (rho_async_contracts), [2](#)
RhoPromiseTask (rho_async_contracts), [2](#)
RhoRejectedTask (rho_async_contracts), [2](#)
RhoSerialQueue (rho_async_contracts), [2](#)
RhoStream (rho_async_contracts), [2](#)
RhoStreamEnd (rho_async_contracts), [2](#)
RhoStreamItem (rho_async_contracts), [2](#)
RhoStreamLike (rho_async_contracts), [2](#)
RhoStreamValue (rho_async_contracts), [2](#)
RhoTask (rho_async_contracts), [2](#)
RhoTaskCallbackBridge
 (rho_async_contracts), [2](#)
RhoTaskCallbackError
 (rho_async_contracts), [2](#)
RhoTaskCallbackNameInUse
 (rho_async_contracts), [2](#)
RhoTaskCallbackRegistrationError
 (rho_async_contracts), [2](#)
RhoTaskCallbackRemovalError
 (rho_async_contracts), [2](#)
RhoTaskQueue (rho_async_contracts), [2](#)
RhoTaskStream (rho_async_contracts), [2](#)
RhoTimeoutError (rho_async_contracts), [2](#)