Package: cozip (via r-universe)

October 31, 2026

Type Package

Title Cloud-Optimized ZIP - Open a ZIP Like a Table

Version 2026.9.27

Date 2026-09-27

Description R bindings to libcozip, the reference library for the Cloud-Optimized ZIP (cozip) format.

License MIT + file LICENSE

URL https://github.com/asterisk-labs/cozip

BugReports https://github.com/asterisk-labs/cozip/issues

Encoding UTF-8

Language en-US

Depends R (>= 4.3)

Imports arrow (>= 14.0.0), bit64, duckdb, DBI, tibble,

Suggests testthat (>= 3.0.0), withr, sf, geoarrow

SystemRequirements GNU make, C++23, pkg-config, libcurl (>= 7.83.0), OpenSSL (>= 3.0)

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libxml2-dev libssl-dev pkg-config

Repository https://asterisk-labs.r-universe.dev

Date/Publication 2026-10-31 02:56:36 UTC

RemoteUrl https://github.com/asterisk-labs/cozip

RemoteRef main

RemoteSha 534fa13ac8dfa167ffbac5a64ab93ec54a9aab9b

RemoteSubdir r

Contents

create 2
read 2
stage_create 3
stage_metadata 4

Index **5**

`create` *Write a cozip archive in one call*

Description

Wraps `stage_metadata()` + `arrow::write_parquet()` + `stage_create()`. For GeoParquet or custom write options, use the staging functions directly.

Usage

```
create(out_path, table, temp_dir = NULL)

write(out_path, table, temp_dir = NULL)
```

Arguments

`out_path`	Destination archive path.
`table`	arrow::Table with `name` and `path`.
`temp_dir`	Directory for the temp metadata parquet (defaults to `tempdir()`).

Value

Absolute path of the created archive.

`read` *Read the manifest of a FLAT-profile cozip archive*

Description

Reads the byte-0 index natively, fetches only the `__metadata__` Parquet manifest, and decodes it with DuckDB, which loads no extension. `name`, `offset`, `size` (and `cozip:location` when `location = TRUE`) are always included; use `columns` to bring extras, `NULL` brings all. Archives using any profile other than Flat (`profile = 1`), including TACO (`profile = 2`), are rejected.

Usage

```
read(source, columns = NULL, location = TRUE)
```

Arguments

source	Local path or http(s)/s3/gcs/azure/hf URL to the .zip.
columns	Character vector of extra columns. NULL returns every column.
location	Include the cozip:location column.

Value

A tibble.

---

stage_create *Pack a cozip archive from sources and a user-written parquet*

Description

The metadata parquet is embedded verbatim, so any GeoParquet or schema metadata you wrote is preserved.

Usage

```
stage_create(out_path, paths, metadata_parquet, validate = TRUE)
```

Arguments

out_path	Destination archive path.
paths	data.frame with name and path in the same row order as metadata_parquet.
metadata_parquet	Path to a parquet with name, offset, size columns and no path column.
validate	If TRUE, re-runs the plan and checks that the parquet matches.

Value

Absolute path of the created archive.

`stage_metadata` *Plan a cozip FLAT archive*

Description

Computes offsets and sizes for the user entries and returns the canonical metadata table plus a row-aligned `paths` data.frame. The function checks each source and reads its file size; it does not read payload contents.

Usage

```
stage_metadata(table)
```

Arguments

`table`	arrow::Table with `name` and `path`. Extras are preserved in the metadata.

Details

If you reorder `metadata` before writing it, reorder `paths` the same way or `stage_create()` will reject the mismatch.

Value

List with `metadata` (arrow::Table) and `paths` (data.frame).

Index

arrow::write_parquet(), *2*

create, 2

read, 2

stage_create, 3
stage_create(), *2*, *4*
stage_metadata, 4
stage_metadata(), *2*

tempdir(), *2*

write *(*create*)*, 2