Package: Ropendal (via r-universe)

June 3, 2026

Type Package

Title Abstract Filesystem Access for R via 'Apache OpenDAL'

Version 0.0.1.9000

Description Provides a byte-oriented abstract filesystem interface for R backed by the Rust crate of 'Apache OpenDAL' <https://opendal.apache.org/>. The package is designed around filesystem primitives, asynchronous Aio-like operations inspired by 'nanonext', pluggable raw-vector serializers, and a native C API for direct async byte access by other R packages.

License GPL (>= 3)

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

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

SystemRequirements Cargo (Rust's package manager), rustc >= 1.88.0, GNU make

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports S7

Suggests tinytest, roxygen2, rmarkdown, knitr, pkgdown, Rtinycc

VignetteBuilder knitr

Config/pak/sysreqs make libclang-dev

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

Date/Publication 2026-06-03 20:29:26 UTC

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

RemoteRef HEAD

RemoteSha b12246ea15fd16cf4d686d9358f5ea7aa153b7f4

Contents

`OpendalAio` *Asynchronous operation handle.*

Description

Asynchronous operation handle.

Usage

```
OpendalAio
```

Format

An object of class `Ropendal::OpendalAio__bundle` (inherits from `savvy_Ropendal__sealed`) of length 0.

`OpendalCredentialProvider`
Explicit credential provider.

Description

Explicit credential provider.

Usage

```
OpendalCredentialProvider
```

Format

An object of class `Ropendal::OpendalCredentialProvider__bundle` (inherits from `savvy_Ropendal__sealed`) of length 5.

OpendalFs — *Filesystem handle backed by Apache OpenDAL.*

Description

Filesystem handle backed by Apache OpenDAL.

Usage

```
OpendalFs
```

Format

An object of class `Ropendal::OpendalFs__bundle` (inherits from `savvy_Ropendal__sealed`) of length 2.

OpendalHttpFixture — *Internal HTTP fixture for tests.*

Description

Internal HTTP fixture for tests.

Usage

```
OpendalHttpFixture
```

Format

An object of class `Ropendal::OpendalHttpFixture__bundle` (inherits from `savvy_Ropendal__sealed`) of length 1.

OpendalLsIter — *Streaming listing iterator over one prefix.*

Description

Streaming listing iterator over one prefix.

Usage

```
OpendalLsIter
```

Format

An object of class `Ropendal::OpendalLsIter__bundle` (inherits from `savvy_Ropendal__sealed`) of length 0.

`OpendalReadIter` *Chunked read iterator over one object.*

Description

Chunked read iterator over one object.

Usage

```
OpendalReadIter
```

Format

An object of class `Ropendal::OpendalReadIter__bundle` (inherits from `savvy_Ropendal__sealed`) of length 0.

`OpendalWriteIter` *Chunked write sink for one object.*

Description

Chunked write sink for one object.

Usage

```
OpendalWriteIter
```

Format

An object of class `Ropendal::OpendalWriteIter__bundle` (inherits from `savvy_Ropendal__sealed`) of length 0.

Ropendal-api *Ropendal filesystem API*

Description

Byte-oriented filesystem operations backed by Apache OpenDAL. The current implementation includes local fs, HTTP, S3-compatible, and Google Drive handles; raw byte operations; metadata and listing where supported; error values; Aio handles for read, write, metadata, and namespace operations; and a pure C API.

Usage

```
opendal(scheme = "fs", ..., root = NULL, config = list(), auth = NULL,
        headers = NULL, runtime = runtime_config(), layers = list())
opendal_uri(uri, headers = NULL, runtime = runtime_config(), layers = list())
credentials_s3(access_key_id, secret_access_key,
               session_token = "", region = "", source = "direct")
credentials_gcs(token = "", service_account_key = "",
                credential_path = "", scope = "", source = "direct")
credentials_azblob(account_name = "", account_key = "", sas_token = "",
                   endpoint = "", source = "direct")
credentials_gdrive(access_token = "", refresh_token = "",
                   client_id = "", client_secret = "", source = "direct")
credentials_gdrive3(secret_json,
                   tokens_json = file.path(dirname(secret_json), "tokens.json"),
                     scope = "https://www.googleapis.com/auth/drive")
credential_schemes(provider)
credential_config(provider, service)
credential_summary(provider)
runtime_config(threads = NULL)
layer_concurrent_limit(max)
layer_timeout(request_timeout = NULL, io_timeout = NULL)
fs_info(fs)
fs_capabilities(fs)
fs_normalize_path(fs, path, directory = FALSE)
opt(fs, name)
opt(fs, name) <- value
serial_config(class, sfunc, ufunc)
codec_config(name, class = "raw", sfunc = NULL, ufunc = NULL)
serialize_raw(x, config = list())
deserialize_raw(x, config = list())
fs_read(fs, path, offset = 0, size = NULL, end = NULL,
        result = c("auto", "flat", "nested"), batch_concurrency = NULL,
        read_concurrency = NULL, chunk_size = NULL, coalesce_gap = NULL,
        mode = c("raw", "serial", "text", "codec"), encoding = "UTF-8",
        serial_config = opt(fs, "serial"), codec = opt(fs, "codec"))
fs_read_aio(fs, path, offset = 0, size = NULL, end = NULL,
```

```
            result = c("auto", "flat", "nested"), batch_concurrency = NULL,
            read_concurrency = NULL, chunk_size = NULL, coalesce_gap = NULL,
            mode = c("raw", "serial", "text", "codec"), encoding = "UTF-8",
            serial_config = opt(fs, "serial"), codec = opt(fs, "codec"))
fs_read_bytes(fs, path, offset = 0, size = NULL, end = NULL,
              result = c("auto", "flat", "nested"), batch_concurrency = NULL,
              read_concurrency = NULL, chunk_size = NULL, coalesce_gap = NULL)
fs_read_bytes_aio(fs, path, offset = 0, size = NULL, end = NULL,
                  result = c("auto", "flat", "nested"), batch_concurrency = NULL,
                  read_concurrency = NULL, chunk_size = NULL, coalesce_gap = NULL)
## S3 method for class 'OpendalBytes'
as.raw(x)
## S3 method for class 'OpendalBytes'
length(x)
fs_read_iter(fs, path, chunk_size, offset = 0, size = NULL,
             read_concurrency = NULL, coalesce_gap = NULL)
read_iter_next(iter)
read_iter_collect(iter)
fs_tell(iter)
fs_seek(iter, offset, whence = c("start", "current", "end"))
fs_write(fs, path, data, batch_concurrency = NULL,
         write_concurrency = NULL, chunk_size = NULL,
         mode = c("raw", "serial", "text", "codec"), encoding = "UTF-8",
         serial_config = opt(fs, "serial"), codec = opt(fs, "codec"))
fs_write_aio(fs, path, data, batch_concurrency = NULL,
             write_concurrency = NULL, chunk_size = NULL,
             mode = c("raw", "serial", "text", "codec"), encoding = "UTF-8",
             serial_config = opt(fs, "serial"), codec = opt(fs, "codec"))
fs_replace(fs, path, data, batch_concurrency = NULL,
           write_concurrency = NULL, chunk_size = NULL,
           mode = c("raw", "serial", "text", "codec"), encoding = "UTF-8",
           serial_config = opt(fs, "serial"), codec = opt(fs, "codec"))
fs_replace_aio(fs, path, data, batch_concurrency = NULL,
               write_concurrency = NULL, chunk_size = NULL,
               mode = c("raw", "serial", "text", "codec"), encoding = "UTF-8",
               serial_config = opt(fs, "serial"), codec = opt(fs, "codec"))
fs_append(fs, path, data, batch_concurrency = NULL,
          write_concurrency = NULL, chunk_size = NULL,
          mode = c("raw", "serial", "text", "codec"), encoding = "UTF-8",
          serial_config = opt(fs, "serial"), codec = opt(fs, "codec"))
fs_append_aio(fs, path, data, batch_concurrency = NULL,
              write_concurrency = NULL, chunk_size = NULL,
              mode = c("raw", "serial", "text", "codec"), encoding = "UTF-8",
              serial_config = opt(fs, "serial"), codec = opt(fs, "codec"))
fs_write_iter(fs, path, create = TRUE, append = FALSE,
              write_concurrency = NULL, chunk_size = NULL)
write_iter_write(iter, data)
write_iter_close(iter)
```

```
fs_stat(fs, path, batch_concurrency = NULL)
fs_stat_aio(fs, path, batch_concurrency = NULL)
fs_stats(fs, path, batch_concurrency = NULL)
fs_stats_aio(fs, path, batch_concurrency = NULL)
fs_exists(fs, path, batch_concurrency = NULL)
fs_exists_aio(fs, path, batch_concurrency = NULL)
fs_ls(fs, path = "", recursive = FALSE, limit = NULL, start_after = NULL)
fs_ls_aio(fs, path = "", recursive = FALSE,
          limit = NULL, start_after = NULL)
fs_ls_iter(fs, path = "", recursive = FALSE, page_size = 1000,
           limit = NULL, start_after = NULL, prefetch = 0)
fs_walk_iter(fs, path = "", page_size = 1000,
             limit = NULL, start_after = NULL, prefetch = 0)
ls_iter_next(iter)
ls_iter_collect(iter)
walk_iter_next(iter)
walk_iter_collect(iter)
fs_mkdir(fs, path)
fs_mkdir_aio(fs, path, batch_concurrency = NULL)
fs_delete(fs, path, recursive = FALSE, batch_concurrency = NULL)
fs_delete_aio(fs, path, recursive = FALSE, batch_concurrency = NULL)
fs_copy(fs, from, to)
fs_copy_aio(fs, from, to, batch_concurrency = NULL)
fs_rename(fs, from, to)
fs_rename_aio(fs, from, to, batch_concurrency = NULL)
collect_aio(aio)
collect_aio_(aio)
call_aio(aio)
call_aio_(aio)
stop_aio(aio)
poll_aio(aio)
cv()
cv_value(cv)
cv_reset(cv)
cv_signal(cv)
cv_wait(cv, interval = 0.001)
cv_until(cv, msec = 0, interval = 0.001)
aio_monitor(aio, cv = cv())
read_monitor(monitor)
race_aio(aio, timeout = NULL, interval = 0.001)
unresolved(x = NULL)
is_error_value(x)
error_kind(x)
error_message(x)
error_operation(x)
error_path(x)
```

Arguments

scheme	OpenDAL service scheme.
...	Named service configuration entries.
root	Root path/prefix for the service.
config	For opendal(), named list of service configuration entries. For serialize_raw() / deserialize_raw(), serialization config.
auth	Explicit credential provider.
headers	Named scalar character list/vector of HTTP headers for http/https filesystems; values may contain credentials and are not printed by Ropendal.
runtime	Runtime configuration from runtime_config().
layers	List of explicit filesystem layers, currently including layer_concurrent_limit() and layer_timeout().
threads	Number of Tokio worker threads for a filesystem handle.
max	Maximum total in-flight backend operations for a filesystem handle.
request_timeout	Operation timeout in seconds for non-streaming backend operations such as stat, exists, delete, copy, rename, and mkdir. If only one timeout field is supplied, the omitted field uses OpenDAL's timeout-layer default.
io_timeout	I/O timeout in seconds for backend read/write streams and listing iteration. If only one timeout field is supplied, the omitted field uses OpenDAL's timeout-layer default.
uri	OpenDAL URI.
access_key_id, secret_access_key	S3-compatible access key fields.
session_token	Optional S3-compatible session token.
region	Optional S3-compatible signing region.
token	Google Cloud Storage OAuth2 bearer token.
service_account_key	Google Cloud Storage service-account JSON string.
credential_path	Explicit path to Google Cloud Storage credential JSON.
account_name	Azure Blob Storage account name.
account_key	Azure Blob Storage account key.
sas_token	Azure Blob Storage SAS token.
endpoint	Optional Azure Blob Storage endpoint.
access_token	Google Drive access token, mutually exclusive with refresh_token.
refresh_token	Google Drive refresh token.
client_id, client_secret	OAuth client fields required with refresh_token.
source	Redacted source label for a credential provider.

secret_json	Path to a JSON file containing Google Drive client_id and client_secret.
tokens_json	Path to a token JSON file containing a refresh token.
scope	Scope used to choose a refresh token from tokens_json.
provider	Credential provider object.
service	OpenDAL service scheme for credential materialization.
fs	Ropendal filesystem handle.
path	Root-relative path or paths.
directory	Whether to normalize as a directory path.
offset	Zero-based byte offset.
size	Number of bytes to read, or NULL to read to EOF.
end	Exclusive byte end offset.
result	Requested result shape.
name	Option name.
value	Option value.
class	Class name or names matched for custom serialization.
sfunc, ufunc	Serializer and deserializer functions for serial_config(). R-closure codecs are not supported; codec transforms are native byte transforms.
mode	Materialization mode: raw bytes, text strings, serialized R objects, or raw bytes passed through an explicit codec.
encoding	Text encoding for mode = "text" reads and writes. Encodings whose encoded bytes contain NUL bytes are rejected; use raw mode for those storage formats.
serial_config	Serialization config, normally opt(fs, "serial").
codec	Optional native byte codec name or codec_config() object, normally opt(fs, "codec").
batch_concurrency	Optional maximum number of independent paths/ranges to process concurrently.
read_concurrency	Optional per-object OpenDAL read concurrency for large reads where the backend supports chunked/concurrent reads.
write_concurrency	Optional per-object OpenDAL write concurrency for large writes where the backend supports multipart/concurrent writes.
chunk_size	Optional read/write chunk size in bytes for OpenDAL's per-object transfer planning.
coalesce_gap	Optional byte gap for coalescing nearby read ranges.
data	Raw vector, OpendalBytes handle, or list mixing raw vectors and OpendalBytes handles for multiple paths in raw mode; arbitrary R object or list of objects in serial mode.
iter	Read, write, listing, or walking iterator handle.
whence	Seek origin for read iterators: iterator start, current, or end.

create	Whether a write iterator should create only and fail if the target exists.
append	Whether a write iterator should append rather than replace.
recursive	Whether to recurse for operations that support it.
limit	Optional maximum number of listing entries to materialize or return across an iterator.
start_after	Optional root-relative listing continuation marker; entries less than or equal to this path are skipped where supported. Iterator $cursor values are last yielded paths and can be used as best-effort start_after markers for lexically ordered listings; they are not opaque backend continuation tokens.
page_size	Maximum number of entries returned by one iterator page.
prefetch	Number of listing entries an iterator may buffer ahead in Rust/Tokio. Positive values may start bounded background prefetch at iterator construction; use 0 to keep strict demand-driven iteration.
from, to	Source and destination paths.
aio	Aio handle.
cv	Condition variable object from cv().
monitor	Monitor handle from aio_monitor().
msec	Milliseconds to wait for cv_until().
timeout	Optional milliseconds to wait for race_aio().
interval	Poll interval in seconds for R-thread wait helpers.
x	Object to inspect.

Details

Filesystem failures are returned as classed error values. Invalid arguments and internal runtime failures may signal R errors. Credential helpers return classed providers with redacted printing; pass them with auth =.

Aio handles expose read-only active bindings. $value returns an unresolvedValue while the operation is pending and the resolved value or error value afterwards. $data and $result are aliases for $value, while $state, $resolved, and $error expose readiness and error inspection. collect_aio() waits and returns the value. call_aio() waits, updates the Aio, and returns the Aio invisibly. unresolved() constructs the unresolved sentinel; unresolved(aio) and unresolved(value) are predicates.

Value

Filesystem handles, raw vectors, OpendalBytes handles, deserialized R objects, metadata lists, logical results, Aio handles, or classed error values depending on the operation and mode.

Index