

Package: vsils (via r-universe)

May 11, 2026

Title Scan 'CRAN' Packages for 'GDAL'-Readable Files

Version 0.1.0

Description Queries a package source tarball directly from CRAN (via 'GDAL' virtual file system) and identifies files that 'GDAL' can read. No package or file download is required.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports digest, dplyr, gdalraster, memoise, purrr, tibble, tools

URL <https://github.com/hypertidy/vsils>

BugReports <https://github.com/hypertidy/vsils/issues>

Suggests curl, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libxml2-dev libzstd-dev

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

Date/Publication 2026-02-03 00:47:28 UTC

RemoteUrl <https://github.com/hypertidy/vsils>

RemoteRef HEAD

RemoteSha 19700093f085659035d1780173c61a9a063cea50

Contents

cran_db_refresh	2
cran_package_db	2
scan_package	3

Index	4
--------------	----------

cran_db_refresh *Refresh a cached CRAN packages DB.*

Description

The vsils function `cran_package_db()` is *memoized*, meaning that its output is cached by the memoise package, with a timeout every hour. Manually bust this cache by running `cran_db_refresh()`.

Usage

```
cran_db_refresh()
```

Value

logical TRUE, or FALSE if no such cache existed

Examples

```
## Not run:  
cran_db_refresh()  
  
## End(Not run)
```

cran_package_db *Get CRAN package database.*

Description

This is a memoized (cached) copy of `tools::CRAN_package_db()`, it refreshes once an hour.

Usage

```
cran_package_db()
```

Details

Force refresh with `cran_db_refresh()`.

See Also

```
cran_db_refresh
```

Examples

```
cran_package_db()
```

scan_package	<i>Scan a CRAN package for GDAL-readable files</i>
--------------	--

Description

Queries a package source tarball directly from CRAN (via `/vsitar//vsicurl/`) and identifies files that GDAL can read. No download required.

Usage

```
scan_package(package)
```

Arguments

`package` Character string, name of a package available on CRAN.

Details

The function reads the package tarball directly from CRAN using GDAL's virtual file system handlers (`/vsitar/` and `/vsicurl/`), so files can be identified without downloading the entire package.

The returned dsn values can be opened directly with GDAL-based tools (e.g. `new(gdalraster::GDALVector,)`, `new(gdalraster::GDALRaster,)`, `terra::vect()`, `terra::rast()`, `sf::read_sf()`).

Some false positives may occur (e.g. CSV files detected as spatial, PNG logos, orphan shapefile sidecars). The XYZ driver in particular may incorrectly match spatial weights files (`.gal`, `.gwt`).

Value

A tibble with columns:

- `package`: the package name
- `driver`: GDAL driver name (e.g. "GTiff", "GPKG", "netCDF")
- `dsn`: full GDAL-readable URI to the file

Returns NULL if no GDAL-readable files are found.

Examples

```
scan_package("gdalraster")

# Scan multiple packages
packages <- c("spData", "gdalraster", "sf", "terra", "stars", "vapour")
results <- purrr::map_dfr(packages, purrr::possibly(scan_package, NULL))

# reference a discovered file directly
gpkg <- dplyr::filter(results, package == "spData", driver == "GPKG") |> dplyr::slice(1L)
v <- new(gdalraster::GDALVector, gpkg$dsn)
v$info()
v$close()
```

Index

`cran_db_refresh`, [2](#)

`cran_package_db`, [2](#)

`scan_package`, [3](#)