Package 'dsn'

July 17, 2026

Title Data Source Name and Description Helpers for Use with 'GDAL'

Version 0.1.0

Description Simple helpers for 'GDAL' data source names ('DSN'), prefix and suffix and other handling. 'GDAL' is the Geospatial Data Abstraction Library <https://gdal.org/>, not used by this package directly.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/hypertidy/dsn, https://hypertidy.github.io/dsn/

BugReports https://github.com/hypertidy/dsn/issues

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Michael Sumner [aut, cre],
Matt Dowle [ctb] (original idea for C_addr, from data.table::address)

Maintainer Michael Sumner <mdsumner@gmail.com>

Repository CRAN

Date/Publication 2026-07-17 12:50:24 UTC

Contents

datatype *Return the type name of the GDAL data type.*

Description

Return the type name of the GDAL data type.

Usage

```
gdal_datatypes()

datatype(x)
```

Arguments

x	integer as returned by GDAL, or osgeo.gdal.Open().GetDatatype()

Value

character string of the type name (the name of the constant in GDAL, e.g. GDT_Byte)

Examples

```
datatype(1)
names(gdal_datatypes())
```

gcp_extent *Create a set of GCPs (ground control points) from dimension, extent*

Description

Create a set of GCPs (ground control points) from dimension, extent

Usage

```
gcp_extent(dimension, extent = NULL)

gcp_extent_arg(gcp)
```

Arguments

dimension	size of grid 'ncol,nrow'
extent	extent 'xmin,xmax,ymin,ymax'
gcp	ground control point ('col,row,x,y')

Value

gcp_extent returns the col,row,x,y values, gcp_extent_arg returns formatted as a GDAL 'vrt://' connection string

Examples

```
gcp_extent(c(10, 20))
dsn <- sprintf("vrt://%s?%s", mem(volcano), gcp_extent_arg(gcp_extent(dim(volcano))))
gcp <- gcp_extent(dim(volcano), c(-180, 180, -90, 90))
gcp_extent_arg(gcp)
```

---

mem *Generate a data source name (DSN) for the GDAL MEM driver*

Description

An array in memory can be referenced by a GDAL data source. The DSN points directly at the memory of x, no copy is made.

Usage

```
mem(
  x,
  extent = NULL,
  projection = "",
  dimension = NULL,
  PIXELOFFSET = NULL,
  LINEOFFSET = NULL,
  BANDOFFSET = NULL
)
```

Arguments

x	an R array of numeric (double) type, see Details
extent	optional extent of the data in x,y c(xmin, xmax, ymin, ymax)
projection	projection string (optional, sets the SPATIALREFERENCE of the MEM driver since GDAL 3.7)
dimension	size in pixels c(PIXELS, LINES), defaults to dim(x), see Details
PIXELOFFSET	pixel offset
LINEOFFSET	line offset
BANDOFFSET	band offset

Details

The DSN is only valid while `x` is alive and unmodified: keep a reference to `x` for as long as the DSN is in use, or R's garbage collector may free or reuse that memory. For the same reason `x` must already be a double array (`storage.mode(x) <- "double"` to convert beforehand); `mem()` will not convert it for you, because converting makes a temporary copy whose memory does not survive this function returning.

This DSN only works in the current R session, with GDAL read and query tools (terra, sf, gdalcubes, vapour, gdalraster, etc.). Since GDAL 3.10, opening a `MEM:::DATAPOINTER` DSN is disabled by default for security reasons: set the configuration option `GDAL_MEM_ENABLE_OPEN=YES` (e.g. `Sys.setenv(GDAL_MEM_ENABLE_OPEN = "YES")`) to allow it.

`dimension` defaults to `dim(x)`; for an R matrix that is `(nrow, ncol)` used as `(PIXELS, LINES)`, so GDAL scanlines run down the columns of the R matrix (memory order is preserved, orientation is transposed relative to the on-screen orientation of `graphics::image()`).

Value

character string, a DSN for use by GDAL

Examples

```
mem(volcano)

m <- matrix(c(0, 0, 0, 1), 5L, 4L)
mem(m)
```

`prefix` *Prefix handlers for GDAL data source names*

Description

Add required prefixes, or remove them. The `/vsiXXX/` prefixes chain, exactly as GDAL's virtual filesystem chains, so they compose by nesting, e.g. `vsizip(vsicurl(x))` gives `"/vsizip//vsicurl/<x>"`.

Usage

```
vsicurl(x, sign = FALSE)

vsizip(x)

vsis3(x)

vsitar(x)

vsigzip(x)

driver(x, driver = "")
```

```
netcdf(x)

unprefix(x)

unvsicurl(x)
```

Arguments

x	character vector, of data source names (file paths, urls, database connection strings, or GDAL dsn)
sign	configure for automatic Planetary Computer signing by GDAL
driver	character vector of appropriate GDAL driver name

Value

character vector

Examples

```
vsicurl("https://netcdf-r-us.org/f.nc")

driver("somefile.h5", "HDF5")

unvsicurl("/vsicurl/https://netcdf-r-us.org/f.nc")

unprefix("NETCDF:/u/user/somefile.nc")

## chaining virtual filesystems
vsizip(vsicurl("https://host/data.zip/inner/layer.shp"))

## MPC signing
mpc <- "https://sentinel2l2a01.blob.core.windows.net/sentinel2-l2/.../T43DFE_B04_10m.tif"
vsicurl(mpc , sign = TRUE)
```

---

sds *Wrapping handlers for GDAL data source names*

Description

Subdataset and VRT connection strings.

Usage

```
sds(x, varname, driver, quote = TRUE)
```

Arguments

x	character vector, of data source names (file paths, urls, database connection strings, or GDAL dsn)
varname	named of variable in DSN
driver	driver to use, e.g. "NETCDF", "HDF5"
quote	wrap the core dsn in escaped double quotes, or not

Value

character string of the form "DRIVER:%s:varname"

Examples

```
f <- "myfile.nc"
sds(f, "variable", "NETCDF", quote = FALSE)
```

---

vrtcon *VRT connection*

Description

Create a vrt connection from an input string and named arguments.

Usage

```
vrtcon(x, ...)
```

Arguments

x	character vector, of data source names (file paths, urls, database connection strings, or GDAL dsn)
...	named arguments like 'a_srs="OGC:CRS84"

Details

As of writing (GDAL 3.7.0DEV 2022-12-12) the only available named arguments are 'a_srs', 'bands', 'a_ullr' but that doesn't stop this function.

Value

character string in the form "vrt://%s?arg1&arg2"

Examples

```
vrtcon("myfile.nc", a_ullr = "0,90,360,-90", bands="1,2,1")
```

Index