Package: wince (via r-universe)

August 16, 2024

Title Write NetCDF Files for the Obvious Situations

Version 0.0.0.9002

Description Create a 3D 'NetCDF' with time or depth as the third
dimension, and convention-conforming coordinate system
metadata. Eventually we'll set this up to write slice by slice
which is easy once the file is instantiated, but atm it's
purely an experiment. We might include capability to create 4D
ones too.

License MIT + file LICENSE

Imports dplyr, ncdf4, RNetCDF, vaster

Encoding UTF-8

Language es

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/hypertidy/wince

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

Remotes hypertidy/vaster

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

RemoteUrl https://github.com/hypertidy/wince

RemoteRef HEAD

RemoteSha 6baf4c388f70873ab3efa91dde12f3df0540c7b7

Contents

---

add_grid_mapping   *Add grid mapping*

Description

grid_mapping by CF convention

Usage

```
add_grid_mapping(x, mapping, overwrite = FALSE)
```

Arguments

x	file name
mapping	list of name of variable and attributes for mapping params (see DEtails)
overwrite	enforce user override to actually update the file

Details

mapping is a list with elements 'name' of the crs variable, and 'atts' list with named attributes and values (e.g. standard_parallel = c(10, 20))

Value

the name of the file modified, returned invisibly

Examples

```
mapping <- list(name = "crs",
                atts = list(grid_mapping_name = "lambert_conformal_conic",
                            standard_parallel = c(-10, -50),
                            #standard_parallel = -50,
                            latitude_of_projection_origin = -30,
                            longitude_of_central_meridian =  134.33,
                            semi_major_axis = 6370000))
```

write_nc *Create raster file in NetCDF.*

Description

Rasters need extent, dimension, projection - this writer gives the correct metadata for simple raster inputs with minimal generality and no fuss.

Usage

```
write_nc(
  data,
  filename = NULL,
  extent = NULL,
  title = "raster",
  zvar = NULL,
  z_type = NULL,
  data_name = NULL,
  data_unit = "some.unit",
  long_name = "some.long.name",
  params = "",
  overwrite = FALSE
)
```

Arguments

filename	the NetCDF file to create
title	name of the model
zvar	actual time steps, need to be regularly space
overwrite	set to TRUE to clobber an existing file
transp_params	optional details to put in the NetCDF notes

Details

These functions aim to minimize the amount of manual handling of details, creating an NetCDF file that can be modified directly.

Value

the filename of the output (use ncdf4 to inspect, modify it)

Examples

```
png <- system.file("textures/world.png", package = "rgl", mustWork = TRUE)
arr <- aperm(png::readPNG(png), c(2, 1, 3))
arr <- arr[,ncol(arr):1, ]
f <- write_nc(arr, extent = c(-180, 180, -90, 90), data_name = "world_image")
```

```
# terra::plotRGB(terra::rast(f) * 256)
# maps::map(add = T)
```

Index