Package: rgeopackage (via r-universe)

November 4, 2024

Title Helper Tools in Creating or Handling GeoPackage Files

Version 0.0.0.900

Description The aim of this R package is to provide helper tools in creating or handling GeoPackage files, in order to complement other R spatial packages or GDAL. The package has no dependencies on other spatial packages and is not tied to any particular package by design.

License GPL (>= 3)

Depends R (>= 3.1.0)

Suggests covr, dplyr, openssl, RSQLite, sf, stars, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

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

RemoteUrl https://github.com/r-spatial/rgeopackage

RemoteRef HEAD

RemoteSha 1ea3d08fc22eeb1e7109a090643cd90354c74f2e

Contents

amend_timestamp *Amend the timestamp(s) in a GeoPackage file*

Description

Overwrites all timestamps (column `last_change`) of the `gpkg_contents` table in an existing GeoPackage file. If the optional table `gpkg_metadata_reference` is present, does the same with its `timestamp` column. As such the function assists in making a binary-reproducible GeoPackage file.

Usage

```
amend_timestamp(dsn, timestamp = Sys.time(), verbose = TRUE)
```

Arguments

`dsn`	the path to the GeoPackage file (*.gpkg)
`timestamp`	a `Date` or `POSIXct` object, used to generate the timestamp. For a `Date` object, time will be considered as `00:00:00 UTC`. Defaults to system time, however must be set explicitly for reproducible workflows.
`verbose`	Logical. For each relevant table, prints a message with the number of affected rows.

Details

Internally the timestamp is converted to a specific ISO 8601 format that is required by the GeoPackage standard.

Value

`NULL` is returned invisibly.

Note

Directly editing a GeoPackage is not advised; whenever possible use `preset_timestamp()` since it goes via GDAL.

However `amend_timestamp()` is especially useful when a GeoPackage file also contains a timestamp in the optional table `gpkg_metadata_reference`, as GDAL does not control that timestamp as of writing (for GDAL 3.1.3). See a corresponding issue in the GDAL source repository.

Author(s)

Floris Vanderhaeghe, https://github.com/florisvdh

See Also

Other functions to control the GeoPackage timestamp(s): `preset_timestamp`, `sf::st_write`

Examples

```
library(sf)
library(openssl)
md5sum <- function(x) paste(md5(file(x)))

# Using existing geopackage with vector layer:
filepath <- system.file("gpkg/b_pump.gpkg", package = "sf")
(md5_original <- md5sum(filepath))

sf_layer <- read_sf(system.file("gpkg/b_pump.gpkg", package = "sf"))

# A rewrite changes the checksum:
filepath_notimeset <- file.path(tempdir(), "b_pump_notimeset.gpkg")
# write 1:
st_write(sf_layer, dsn = filepath_notimeset, delete_dsn = TRUE)
(md5_notimeset1 <- md5sum(filepath_notimeset))
# write 2:
st_write(sf_layer, dsn = filepath_notimeset, delete_dsn = TRUE)
(md5_notimeset2 <- md5sum(filepath_notimeset))
# compare:
md5_notimeset1 == md5_notimeset2

# Setting a fixed date
filepath_timeset <- file.path(tempdir(), "b_pump_timeset.gpkg")
(fixed_date <- as.Date("2020-12-25"))
# write 1 (date):
st_write(sf_layer, dsn = filepath_timeset, delete_dsn = TRUE)
amend_timestamp(filepath_timeset, fixed_date)
md5_timeset1 <- md5sum(filepath_timeset)
# write 2 (date):
st_write(sf_layer, dsn = filepath_timeset, delete_dsn = TRUE)
amend_timestamp(filepath_timeset, fixed_date)
md5_timeset2 <- md5sum(filepath_timeset)
# compare:
all.equal(md5_timeset1, md5_timeset2)

# Setting a fixed time
(fixed_time <- as.POSIXct("2020-12-25 12:00:00", tz = "CET"))
# write 3 (time):
st_write(sf_layer, dsn = filepath_timeset, delete_dsn = TRUE)
amend_timestamp(filepath_timeset, fixed_time)
md5_timeset3 <- md5sum(filepath_timeset)
# write 4 (time):
st_write(sf_layer, dsn = filepath_timeset, delete_dsn = TRUE)
amend_timestamp(filepath_timeset, fixed_time)
md5_timeset4 <- md5sum(filepath_timeset)
# compare:
all.equal(md5_timeset3, md5_timeset4)

# Also works for GPKG 2D gridded coverage (with stars):
library(stars)
library(dplyr)
```

```
filepath_stars <- file.path(tempdir(), "stars_2d.gpkg")

stars_2d <-
  system.file("tif/L7_ETMs.tif", package = "stars") %>%
  read_stars() %>%
  slice(band, 1)
# write 1:
stars_2d %>%
  write_stars(filepath_stars, driver = "GPKG")
amend_timestamp(filepath_stars, fixed_time)
md5_stars1 <- md5sum(filepath_stars)
# write 2:
stars_2d %>%
  write_stars(filepath_stars, driver = "GPKG")
amend_timestamp(filepath_stars, fixed_time)
md5_stars2 <- md5sum(filepath_stars)
# compare:
all.equal(md5_stars1, md5_stars2)
```

`preset_timestamp` *Preset timestamp to reproducibly write GeoPackage files*

Description

Presets the timestamp for usage by GDAL by setting the environment variable `OGR_CURRENT_DATE`. After this, newly written GeoPackage files created by the GDAL vector or raster driver (e.g. through `sf::st_write()` or `stars::write_stars()`) will carry this timestamp. As such `preset_timestamp()` assists in making a binary-reproducible GeoPackage file.

`unset_timestamp()` removes `OGR_CURRENT_DATE` from the environment.

Usage

```
preset_timestamp(timestamp)

unset_timestamp()
```

Arguments

`timestamp`	a `Date` or `POSIXct` object, used to generate the timestamp. For a `Date` object, time will be considered as `00:00:00 UTC`.

Details

The function converts the timestamp to a very specific ISO 8601 format that is required by the GeoPackage standard, including conversion to UTC. Cf. Requirement 15 in version 1.3. GDAL uses the timestamp to set the `last_change` column of the `gpkg_contents` table in newly written GeoPackage files.

The timestamp set by `preset_timestamp()` is adopted by GDAL during the entire session, unless `unset_timestamp()` is called.

Value

Previous value of environment variable `OGR_CURRENT_DATE` is returned invisibly.

Author(s)

Floris Vanderhaeghe, https://github.com/florisvdh

See Also

Other functions to control the GeoPackage timestamp(s): `amend_timestamp`, `sf::st_write`

Examples

```
library(sf)
library(openssl)
md5sum <- function(x) paste(md5(file(x)))

# Using existing geopackage with vector layer:
filepath <- system.file("gpkg/b_pump.gpkg", package = "sf")
(md5_original <- md5sum(filepath))

sf_layer <- read_sf(system.file("gpkg/b_pump.gpkg", package = "sf"))

# A rewrite changes the checksum:
filepath_notimeset <- file.path(tempdir(), "b_pump_notimeset.gpkg")
# write 1:
st_write(sf_layer, dsn = filepath_notimeset, delete_dsn = TRUE)
(md5_notimeset1 <- md5sum(filepath_notimeset))
# write 2:
st_write(sf_layer, dsn = filepath_notimeset, delete_dsn = TRUE)
(md5_notimeset2 <- md5sum(filepath_notimeset))
# compare:
md5_notimeset1 == md5_notimeset2

# Setting a fixed date
filepath_timeset <- file.path(tempdir(), "b_pump_timeset.gpkg")
(fixed_date <- as.Date("2020-12-25"))
preset_timestamp(fixed_date)
# write 1 (date):
st_write(sf_layer, dsn = filepath_timeset, delete_dsn = TRUE)
md5_timeset1 <- md5sum(filepath_timeset)
# write 2 (date):
st_write(sf_layer, dsn = filepath_timeset, delete_dsn = TRUE)
md5_timeset2 <- md5sum(filepath_timeset)
# compare:
all.equal(md5_timeset1, md5_timeset2)

# Setting a fixed time
```

```
(fixed_time <- as.POSIXct("2020-12-25 12:00:00", tz = "CET"))
preset_timestamp(fixed_time)
# write 3 (time):
st_write(sf_layer, dsn = filepath_timeset, delete_dsn = TRUE)
md5_timeset3 <- md5sum(filepath_timeset)
# write 4 (time):
st_write(sf_layer, dsn = filepath_timeset, delete_dsn = TRUE)
md5_timeset4 <- md5sum(filepath_timeset)
# compare:
all.equal(md5_timeset3, md5_timeset4)

# Also works for GPKG 2D gridded coverage (with stars):
library(stars)
library(dplyr)

filepath_stars <- file.path(tempdir(), "stars_2d.gpkg")
(fixed_time <- as.POSIXct("2010-02-14 12:00:00", tz = "CET"))
preset_timestamp(fixed_time)

stars_2d <-
  system.file("tif/L7_ETMs.tif", package = "stars") %>%
  read_stars() %>%
  slice(band, 1)
# write 1:
stars_2d %>%
  write_stars(filepath_stars, driver = "GPKG")
md5_stars1 <- md5sum(filepath_stars)
# write 2:
stars_2d %>%
  write_stars(filepath_stars, driver = "GPKG")
md5_stars2 <- md5sum(filepath_stars)
# compare:
all.equal(md5_stars1, md5_stars2)
```

Index