Package 'CopernicusMarine'

October 13, 2026

Type Package

Title Search Download and Handle Data from Copernicus Marine Service Information

Version 0.5.0

Author Pepijn de Vries [aut, cre, dtc] (ORCID: <https://orcid.org/0000-0002-7961-6646>)

Maintainer Pepijn de Vries <pepijn.devries@outlook.com>

Description Subset and download data from EU Copernicus Marine Service Information: <https://data.marine.copernicus.eu>. Import data on the oceans physical and biogeochemical state from Copernicus into R without the need of external software.

Depends R (>= 4.1.0)

Imports cli, dplyr, httr2 (>= 1.1.0), jsonlite, leaflet, lubridate, purrr, rlang, sf (>= 1.0-24), stars (>= 0.7-1), stringr, tibble, tidyr, xml2

Suggests CFtime (>= 1.7.1), clipr, curl, DT, knitr, lifecycle, ncmeta, ncdf4, paws.storage, paws.common (>= 0.8.0), rmarkdown, units, testthat (>= 3.0.0)

SystemRequirements Requires GDAL >= 3.11 with BLOSC support See <https://pepijn-devries.github.io/CopernicusMarine/articles/blosc.html>

URL https://github.com/pepijn-devries/CopernicusMarine, https://pepijn-devries.github.io/CopernicusMarine/

BugReports https://github.com/pepijn-devries/CopernicusMarine/issues

License GPL (>= 3)

Encoding UTF-8

Config/testthat/edition 3

Collate 'CopernicusMarine-package.r' 'cms_catalogue_entry.R' 'cms_cite_product.r' 'init.R' 'generics.r' 'cms_download_native.R' 'cms_login.r' 'cms_download_subset.r' 'cms_glossary.R' 'cms_ncdf.R' 'cms_product_details.r' 'cms_product_metadata.r' 'cms_product_services.r' 'cms_products_list.r' 'cms_wmts.r' 'import.r' 'translate.R'

Language en-GB

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2026-10-13 07:40:02 UTC

Contents

cms_cite_product 2
cms_download_native 3
cms_download_subset 5
cms_get_client_info 7
cms_get_username 8
cms_glossary 9
cms_login 9
cms_native_proxy 10
cms_native_s3 12
cms_products_list 14
cms_product_details 15
cms_product_metadata 16
cms_product_services 17
cms_translate 18
cms_wmts_details 19
cms_write_ncdf 21
cms_zarr_proxy 22

Index **24**

`cms_cite_product` *How to Cite a Copernicus Marine Product*

Description

[Stable] Get details for properly citing a Copernicus product.

Usage

```
cms_cite_product(product)
```

Arguments

`product`	An identifier (type `character`) of the desired Copernicus marine product. Can be obtained with `cms_products_list`.

Value

Returns a vector of character strings. The first element is always the product title, id and doi. Remaining elements are other associated references. Note that the remaining references are returned as listed at Copernicus. Note that the citing formatting does not appear to be standardised.

Author(s)

Pepijn de Vries

See Also

Other product: `cms_download_native()`, `cms_product_details()`, `cms_product_metadata()`, `cms_product_services()`, `cms_products_list()`

Examples

```
if (interactive()) {
  cms_cite_product("SST_MED_PHY_SUBSKIN_L4_NRT_010_036")
}
```

`cms_download_native` *Download Raw Files as Provided to Copernicus Marine*

Description

[Stable] Full marine data sets can be downloaded using the functions documented here. Use `cms_list_native_files()` to list available files, and `cms_download_native()` to download specific files. Files are usually organised per product, layer, year, month and day.

Usage

```
cms_download_native(
  destination,
  product,
  layer,
  pattern,
  prefix,
  progress = TRUE,
  ...,
  username = cms_get_username(),
  password = cms_get_password()
)

cms_list_native_files(product, layer, pattern, prefix, max = Inf, ...)
```

Arguments

destination	Path where to store the downloaded file(s).
product	An identifier (type character) of the desired Copernicus marine product. Can be obtained with cms_products_list.
layer	The name of a desired layer within a product (type character). Can be obtained with cms_product_services (listed as id column).
pattern	A regular expression (regex) pattern. Only paths that match the pattern will be returned. It can be used to select specific files. For instance if pattern = "2022/06/", only files for the year 2022 and the month June will be listed (assuming that the file path is structured as such, see examples)
prefix	A character string. A prefix to be added to the search path of the files. Only the matching file (info) is downloaded (generally faster then using pattern)
progress	A logical value. When TRUE a progress bar is shown.
...	Ignored
username	Your Copernicus marine user name. Can be provided with cms_get_username() (default), or as argument here.
password	Your Copernicus marine password. Can be provided as cms_get_password() (default), or as argument here.
max	A maximum number of records to be returned.

Value

Returns NULL invisibly.

Author(s)

Pepijn de Vries

See Also

Other download: cms_download_subset(), cms_native_proxy(), cms_native_s3(), cms_zarr_proxy()

Other product: cms_cite_product(), cms_product_details(), cms_product_metadata(), cms_product_services(), cms_products_list()

Examples

```
if (interactive() && requireNamespace("paws.storage")) {
  cms_list_native_files(
    product       = "GLOBAL_ANALYSISFORECAST_PHY_001_024",
    layer         = "cmems_mod_glo_phy_anfc_0.083deg_PT1H-m",
    prefix        = "2022/06/"
  )

## If you omit the prefix, you may want to limit the
## number of results by setting `max`
  cms_list_native_files(
    product       = "GLOBAL_ANALYSISFORECAST_PHY_001_024",
```

```
    layer         = "cmems_mod_glo_phy_anfc_0.083deg_PT1H-m",
    max           = 10
  )

## Prefix can be omitted when not relevant:
  cms_list_native_files(product = "SEALEVEL_GLO_PHY_MDT_008_063")

## Use 'pattern' to download a file for a specific day:
  cms_download_native(
    destination   = tempdir(),
    product       = "GLOBAL_ANALYSISFORECAST_PHY_001_024",
    layer         = "cmems_mod_glo_phy_anfc_0.083deg_PT1H-m",
    prefix        = "2022/06/",
    pattern       = "m_20220630"
  )
}
```

`cms_download_subset` *Subset and Download a Specific Marine Product from Copernicus*

Description

[Stable] Subset and download a specific marine product from Copernicus.

Usage

```
cms_download_subset(
  product,
  layer,
  variable,
  region,
  timerange,
  verticalrange,
  progress = TRUE,
  asset,
  ...,
  username = cms_get_username(),
  password = cms_get_password()
)
```

Arguments

`product`	An identifier (type `character`) of the desired Copernicus marine product. Can be obtained with `cms_products_list`.
`layer`	The name of a desired layer within a product (type `character`). Can be obtained with `cms_product_services` (listed as `id` column).
`variable`	The name of a desired variable in a specific layer of a product (type `character`). Can be obtained with `cms_product_details`. When omitted, all variables are selected.

region	Specification of the bounding box as a vector of numerics WGS84 lat and lon coordinates. Should be in the order of: xmin, ymin, xmax, ymax. When omitted, the entire available region is selected.
timerange	A vector with two elements (lower and upper value) for a requested time range. The vector should be coercible to POSIXct. When omitted, the full available time range is selected.
verticalrange	A vector with two elements (minimum and maximum) numerical values for the depth of the vertical layers (if any). Note that values below the sea surface needs to be specified as negative values. When omitted, the entire available vertical range is selected.
progress	A logical value. When TRUE (default) progress is reported to the console. Otherwise, this function will silently proceed.
asset	Type of asset to be used when subsetting data. Should be one of "default", "ARCO", "static", "omi", "downsampled4", "timeChunked", or "geoChunked". When missing, set to NULL or set to "default", it will use the first asset available for the requested product and layer, in the order as listed before.
...	Ignored (reserved for future features).
username	Your Copernicus marine user name. Can be provided with cms_get_username() (default), or as argument here.
password	Your Copernicus marine password. Can be provided as cms_get_password() (default), or as argument here.

Value

Returns a stars::st_as_stars() object. You can save it to a file with cms_write_ncdf()

Author(s)

Pepijn de Vries

See Also

Other download: cms_download_native(), cms_native_proxy(), cms_native_s3(), cms_zarr_proxy()

Examples

```
if (interactive()) {

  mydata <- cms_download_subset(
    product       = "GLOBAL_ANALYSISFORECAST_PHY_001_024",
    layer         = "cmems_mod_glo_phy-cur_anfc_0.083deg_P1D-m",
    variable      = c("uo", "vo"),
    region        = c(-1, 50, 10, 55),
    timerange     = c("2025-01-01 UTC", "2025-01-02 UTC"),
    verticalrange = c(0, -2)
  )

  plot(mydata["vo"])
```

```
} else {
  message("Make sure to run this in an interactive environment")
}
```

`cms_get_client_info` *Get Information About Copernicus Marine Clients*

Description

[Stable] This function retrieves the client information from the Copernicus Marine Service. Among others, it lists where to find the catalogues required by this package

Usage

```
cms_get_client_info(...)
```

Arguments

`...`	Ignored

Value

In case of success it returns a named `list` with information about the available Copernicus Marine clients.

Author(s)

Pepijn de Vries

See Also

Other supporting: `cms_glossary()`, `cms_translate()`, `cms_write_ncdf()`

Examples

```
if (interactive()) {
  cms_get_client_info()
}
```

`cms_get_username` *Set or get Copernicus Account Details*

Description

[Stable] Set or get username and password throughout an R session. This can be used to obscure your account details in an R script and store them as either an R option or system environment variable.

Usage

```
cms_get_username()

cms_get_password()

cms_set_username(username, method = c("option", "sysenv"))

cms_set_password(password, method = c("option", "sysenv"))
```

Arguments

`username`	Your Copernicus Marine username
`method`	Either `"option"` to use R options to store account details. Use `"sysenv"` to store account details as system environment variable.
`password`	Your Copernicus Marine password

Value

Returns your account details for the `get` variant or nothing in case of the `set` variant.

Author(s)

Pepijn de Vries

See Also

Other authentication: `cms_login()`

Examples

```
if (interactive()) {
  ## Returns your account details only if they have been set for your session
  cms_get_username()
  cms_get_password()
}
```

cms_glossary *Get Copernicus Marine Terminology Glossary*

Description

[Stable] Function that returns a `data.frame` with a glossary of terminology used by the Copernicus Marine Data Service. It is the same `data.frame` that is used to render `vignette("glossary")`.

Usage

```
cms_glossary(search, match_fun = agrep, ...)
```

Arguments

`search`	Search terms to look for in the glossary `data.frame`. Only rows that match these terms are returned. If missing, the entire `data.frame` is returned.
`match_fun`	Function used to filter the `data.frame`. It needs to be a function that uses a `pattern` argument to match the text in the `data.frame` against. It should return a vector of `logical` values or a vector of `integer` row index values. By default it uses `agrepl()`, for a fuzzy match.
`...`	Arguments passed to `match_fun`.

Value

Returns a `data.frame` with glossary info.

See Also

Other supporting: `cms_get_client_info()`, `cms_translate()`, `cms_write_ncdf()`

Examples

```
cms_glossary("variable", ignore.case = TRUE)
```

cms_login *Contact Copernicus Marine Login Page*

Description

[Stable] Contact Copernicus Marine login page and check if login is successful.

Usage

```
cms_login(username = cms_get_username(), password = cms_get_password())
```

Arguments

username	Your Copernicus marine user name. Can be provided with cms_get_username() (default), or as argument here.
password	Your Copernicus marine password. Can be provided as cms_get_password() (default), or as argument here.

Details

This function will return your account details if successful.

Value

Returns a named list with your account details if successful, returns NULL otherwise.

Author(s)

Pepijn de Vries

See Also

Other authentication: cms_get_username()

Examples

```
if (interactive()) {
  cms_login()
}
```

---

cms_native_proxy *Get a Proxy 'stars' Object from a Native Service*

Description

[Stable] The advantage of stars_proxy objects, is that they do not contain any data. They are therefore fast to handle and consume only limited memory. You can still manipulate the object lazily (like selecting slices). These operation are only executed when calling stars::st_as_stars() or plot() on the object.

Usage

```
cms_native_proxy(
  product,
  layer,
  pattern,
  prefix,
  variable,
  ...,
```

```
  username = cms_get_username(),
  password = cms_get_password()
)
```

Arguments

product	An identifier (type character) of the desired Copernicus marine product. Can be obtained with cms_products_list.
layer	The name of a desired layer within a product (type character). Can be obtained with cms_product_services (listed as id column).
pattern	A regular expression (regex) pattern. Only paths that match the pattern will be returned. It can be used to select specific files. For instance if pattern = "2022/06/", only files for the year 2022 and the month June will be listed (assuming that the file path is structured as such, see examples)
prefix	A character string. A prefix to be added to the search path of the files. Only the matching file (info) is downloaded (generally faster then using pattern)
variable	The variable name for which to create the stars_proxy. If omitted it will include all variables in the layer.
...	Ignored
username	Your Copernicus marine user name. Can be provided with cms_get_username() (default), or as argument here.
password	Your Copernicus marine password. Can be provided as cms_get_password() (default), or as argument here.

Details

For more details see vignette("proxy").

Value

A stars_proxy object

Author(s)

Pepijn de Vries

See Also

Other download: cms_download_native(), cms_download_subset(), cms_native_s3(), cms_zarr_proxy()

Examples

```
if (interactive() && requireNamespace("paws.storage")) {
  native_proxy <-
    cms_native_proxy(
      product       = "GLOBAL_ANALYSISFORECAST_PHY_001_024",
      layer         = "cmems_mod_glo_phy_anfc_0.083deg_PT1H-m",
      prefix        = "2022/06/",
```

```
    pattern       = "20220621"
  )
plot(native_proxy["zos", 1:1000, 1:500, 1, 1], axes = TRUE)
}
```

`cms_native_s3` *Navigate Through Copernicus Marine Native File via S3*

Description

[Experimental] Native files (i.e. files as provided by suppliers) are hosted with the Amazon Simple Storage Service (s3). This function generates a `paws.storage::s3()` object, that can be used to navigate and download these files.

Usage

```
cms_native_s3(
  config = list(),
  credentials = list(anonymous = TRUE),
  endpoint = "https://s3.waw3-1.cloudferro.com",
  region = "us-east-1",
  ...
)
```

Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
html

- **use_dual_stack**: Set this to `true` to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`credentials` Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

`endpoint` Optional shorthand for complete URL to use for the constructed client.

`region` Optional shorthand for AWS Region used in instantiating the client.

`...` Ignored

Details

Note that alternative functions in this package provide more convenient routes:

- `cms_list_native_files()`
- `cms_download_native()`
- `cms_native_proxy()`

Value

Returns a `paws.storage::s3()` object, specifically representing the service that hosts Copernicus Marine native data.

See Also

Other download: `cms_download_native()`, `cms_download_subset()`, `cms_native_proxy()`, `cms_zarr_proxy()`

Examples

```
if (interactive() && requireNamespace("paws.storage")) {
  my_s3 <- cms_native_s3()
  my_s3$list_objects_v2("mdl-native-14", MaxKeys = 5)
}
```

cms_products_list *List Products Available from <data.marine.copernicus.eu>*

Description

[Stable] Collect a list of products and some brief descriptions for marine products available from Copernicus. `cms_products_list()` does not use a formal API, but provides a more detailed list. `cms_products_list2()` Does use the formal API, but provides less details.

Usage

```
cms_products_list(..., info_type = c("list", "meta"))

cms_products_list2(...)
```

Arguments

`...`	Allows you to pass (search) query parameters to apply to the list. When omitted, the full list of products is returned.
`info_type`	One of `"list"` (default) or `"meta"`. `"list"` returns the actual list whereas `"meta"` returns meta information for the executed query (e.g. number of hits).

Details

See `vignette("product-info")` for more details.

Value

Returns a `tibble` of products available from https://data.marine.copernicus.eu or a named `list` when `info_type = "meta"`. Returns `NULL` in case on-line services are unavailable.

Author(s)

Pepijn de Vries

See Also

Other product: `cms_cite_product()`, `cms_download_native()`, `cms_product_details()`, `cms_product_metadata()`, `cms_product_services()`

Examples

```
if (interactive()) {
  cms_products_list()

## Query a specific product:
  cms_products_list(freeText = "GLOBAL_ANALYSISFORECAST_PHY_001_024")
}
```

cms_product_details *Obtain Details for a Specific Copernicus Marine Product*

Description

[Stable] Obtain details for a specific Copernicus marine product.

Usage

```
cms_product_details(product, ...)
```

Arguments

product	An identifier (type character) of the desired Copernicus marine product. Can be obtained with cms_products_list.
...	Ignored

Details

See vignette("product-info") for more details.

Value

Returns a named list with product details.

Author(s)

Pepijn de Vries

See Also

Other product: cms_cite_product(), cms_download_native(), cms_product_metadata(), cms_product_services(), cms_products_list()

Examples

```
if (interactive()) {
  cms_product_details("GLOBAL_ANALYSISFORECAST_PHY_001_024")
}
```

`cms_product_metadata` *Obtain Product meta Data*

Description

[Stable] Obtain product meta data such as spatio-temporal bounds of the data.

Usage

```
cms_product_metadata(product, ...)
```

Arguments

`product`	An identifier (type `character`) of the desired Copernicus marine product. Can be obtained with `cms_products_list`.
`...`	Ignored

Details

See `vignette("product-info")` for more details.

Value

Returns a `data.frame`/`tibble` with the metadata. Each row in the `data.frame` represents a layer available for the product.

Author(s)

Pepijn de Vries

See Also

Other product: `cms_cite_product()`, `cms_download_native()`, `cms_product_details()`, `cms_product_services()`, `cms_products_list()`

Examples

```
if (interactive()) {
  cms_product_metadata("GLOBAL_ANALYSISFORECAST_PHY_001_024")
}
```

cms_product_services *Obtain Available Services for a Specific Copernicus Marine Product*

Description

[Stable] Obtain an overview of services provided by Copernicus for a specific marine product.

Usage

```
cms_product_services(product, ...)
```

Arguments

product	An identifier (type character) of the desired Copernicus marine product. Can be obtained with cms_products_list.
...	Ignored.

Details

See vignette("product-info") for more details.

Value

Returns a tibble with a list of available services for a Copernicus marine product.

Author(s)

Pepijn de Vries

See Also

Other product: cms_cite_product(), cms_download_native(), cms_product_details(), cms_product_metadata(), cms_products_list()

Examples

```
if (interactive()) {
  cms_product_services("GLOBAL_ANALYSISFORECAST_PHY_001_024")
}
```

`cms_translate` *Translate Python code or command line request to R list*

Description

[Stable] Use the Copernicus Marine Service website to navigate datasets https://data.marine.copernicus.eu/products. You can specify a query using the website's download form, and copy it's automation download code (either command line or Python) to the system's clipboard. You can then use this function to translate this code to a named list. The list can be used in combination with `cms_download_subset()` to download data. See `vignette("translate")` for more details.

Usage

```
cms_translate(text, ...)
```

Arguments

`text`	The query code as copied from the Copernicus Marine Service website. Both Python and command line code are accepted. When this argument is omitted, the function will look for a query on the system clipboard.
`...`	Ignored

Value

Returns a named list with arguments for `cms_download_subset()`

See Also

Other supporting: `cms_get_client_info()`, `cms_glossary()`, `cms_write_ncdf()`

Examples

```
python_code <-
"import copernicusmarine

copernicusmarine.subset(
  dataset_id=\"cmems_mod_glo_phy_anfc_0.083deg_PT1H-m\",
  variables=[\"uo\",\"vo\"],
  minimum_longitude=-2,
  maximum_longitude=8,
  minimum_latitude=52,
  maximum_latitude=59,
  start_datetime=\"2025-01-01T00:00:00\",
  end_datetime=\"2025-01-01T23:00:00\",
  minimum_depth=0.49402499198913574,
  maximum_depth=0.49402499198913574,
)"

cli_code <-
```

```
"copernicusmarine subset
  --dataset-id cmems_mod_glo_phy_anfc_0.083deg_PT1H-m
  --variable uo
  --variable vo
  --start-datetime 2025-01-01T00:00:00
  --end-datetime 2025-01-01T23:00:00
  --minimum-longitude -2
  --maximum-longitude 8
  --minimum-latitude 52
  --maximum-latitude 59
  --minimum-depth 0.49402499198913574
  --maximum-depth 0.49402499198913574"

if (interactive()) {
  cms_translate(python_code)
  cms_translate(cli_code)
  translated <- cms_translate(cli_code)
  do.call(cms_download_subset, translated)
}
```

`cms_wmts_details` *Obtain a WMTS Entry for Specific Copernicus Marine Products and add to a Leaflet Map*

Description

[Stable] Functions for retrieving Web Map Tile Services information for specific products, layers and variables and add them to a `leaflet` map.

Usage

```
cms_wmts_details(product, layer, variable)

addCmsWMTSTiles(
  map,
  product,
  layer,
  variable,
  tilematrixset = "EPSG:3857",
  time,
  elevation,
  options = leaflet::WMSTileOptions(format = "image/png", transparent = TRUE),
  ...
)

cms_wmts_get_capabilities(
  product,
  layer,
```

```
  variable,
  type = c("data.frame", "xml")
)
```

Arguments

product	An identifier (type character) of the desired Copernicus marine product. Can be obtained with cms_products_list.
layer	The name of a desired layer within a product (type character). Can be obtained with cms_product_services (listed as id column).
variable	The name of a desired variable in a specific layer of a product (type character). Can be obtained with cms_product_details. When omitted, all variables are selected.
map	A map widget object created from leaflet::leaflet()
tilematrixset	A character string representing the tilematrixset to be used. In many cases "EPSG:3857" (Pseudo-Mercator) or "EPSG:4326" (World Geodetic System 1984) are available, but should be checked with cms_wmts_details.
elevation, time	Elevation or time dimension value for which to add the tiles to the map. When missing; or not matching exactly with the values specified by cms_wmts_get_capabilities(); the default dimension value will be used.
options	Passed on to leaflet::addWMSTiles().
...	Passed on to leaflet::addWMSTiles().
type	A character string indicating whether the capabilities should be returned as "data.frame" (default) or "xml" (xml2::xml_new_document()).

Value

cms_wmts_details returns a tibble with details on the WMTS service. cms_wmts_get_capabilities returns either a data.frame or xml_document depending on the value of type. AddCmsWMTSTiles returns a leaflet map updated with the requested tiles.

Author(s)

Pepijn de Vries

Examples

```
if (interactive()) {
  wmts_details <-
    cms_wmts_details(
      product  = "GLOBAL_ANALYSISFORECAST_PHY_001_024",
      layer    = "cmems_mod_glo_phy-thetao_anfc_0.083deg_P1D-m",
      variable = "thetao"
    )

  capabilities <-
    cms_wmts_get_capabilities("GLOBAL_ANALYSISFORECAST_PHY_001_024")
```

```
  if (nrow(wmts_details) > 0) {
    leaflet::leaflet() |>
      leaflet::setView(lng = 3, lat = 54, zoom = 4) |>
      leaflet::addProviderTiles("Esri.WorldImagery") |>
      addCmsWMTSTiles(
        product  = "GLOBAL_ANALYSISFORECAST_PHY_001_024",
        layer    = "cmems_mod_glo_phy-thetao_anfc_0.083deg_P1D-m",
        variable = "thetao")
  }
}
```

`cms_write_ncdf` *Store Typical Copernicus Marine Rasters Objects as NCDF*

Description

[Experimental] The `cms_download_subset()` returns a `stars` class object. This is fine if you want to use it directly in R. But if you want to open it in external software, you need a more acceptable exchange format. You can use this function to store the `stars` object as a NetCDF file.

Usage

```
cms_write_ncdf(x, file, missval = -999, prec = "double", ...)
```

Arguments

`x`	A `stars` class object created with `cms_download_subset()`. Note that this is not a highly generic NetCDF writer, so it makes certain assumptions about the `stars` object, that may not be true for all.
`file`	File path where to store the object.
`missval`	Value used to represent missing data (`NA`) in your object.
`prec`	Precision used for storing the data (`"double"` by default).
`...`	Ignored.

Details

Note that dimensions returned by `cms_download_subset()` have interval values, describing between what range the raster cell lies. When reading a file stored with `cms_write_ncdf()`, you must specify where from the file the bounds of the dimension should be read. Otherwise the GDAL driver will (often incorrectly) deduce dimension values. The example shows how to read the written file using `stars::read_mdim()`.

The `cms_write_ncdf()` function is tailored to `stars` objects returned by `cms_download_subset()`. It might work on other `stars` objects, but it is not guaranteed.

Writing multidimensional data to a standardised format is the source of many headaches. The current implementation is not ideal, which is why it is currently experimental. Perhaps future GDAL release will have better support for writing higher dimension raster data. To be safe, you can also save your data as `.rdata`, or reduce the dimensions by storing specific slices.

Value

Returns NULL invisibly.

See Also

Other supporting: cms_get_client_info(), cms_glossary(), cms_translate()

Examples

```
if (interactive()) {
  ## Download some data to store
  mydata <- cms_download_subset(
    product       = "GLOBAL_ANALYSISFORECAST_PHY_001_024",
    layer         = "cmems_mod_glo_phy-cur_anfc_0.083deg_P1D-m",
    variable      = c("uo", "vo"),
    region        = c(-1, 50, 10, 55),
    timerange     = c("2025-01-01 UTC", "2025-01-02 UTC"),
    verticalrange = c(0, -2)
  )
  fn <- tempfile(fileext = ".nc")
  cms_write_ncdf(mydata, fn)
  mydata2 <- stars::read_mdim(
    fn,
    ## You explicitly need to specify dimension bounds when reading back the data
    bounds = c(
      longitude = "longitude_bnds",
      latitude  = "latitude_bnds",
      elevation = "elevation_bnds"))

  ## clean up after our selves
  unlink(fn, TRUE, TRUE)
}
```

---

cms_zarr_proxy *Get a Proxy 'stars' Object from a Zarr Service*

Description

[Stable] The advantage of stars_proxy objects, is that they do not contain any data. They are therefore fast to handle and consume only limited memory. You can still manipulate the object lazily (like selecting slices). These operation are only executed when calling stars::st_as_stars() or plot() on the object.

Usage

```
cms_zarr_proxy(
  product,
  layer,
  variable,
```

```
  asset,
  ...,
  username = cms_get_username(),
  password = cms_get_password()
)
```

Arguments

product	An identifier (type character) of the desired Copernicus marine product. Can be obtained with cms_products_list.
layer	The name of a desired layer within a product (type character). Can be obtained with cms_product_services (listed as id column).
variable	The name of a desired variable in a specific layer of a product (type character). Can be obtained with cms_product_details. When omitted, all variables are selected.
asset	An asset that is available for the product. Should be one of "native", "wmts", "timeChunked", "downsampled4", or "geoChunked".
...	Ignored (reserved for future features).
username	Your Copernicus marine user name. Can be provided with cms_get_username() (default), or as argument here.
password	Your Copernicus marine password. Can be provided as cms_get_password() (default), or as argument here.

Details

For more details see vignette("proxy").

Value

A stars_proxy object

Author(s)

Pepijn de Vries

See Also

Other download: cms_download_native(), cms_download_subset(), cms_native_proxy(), cms_native_s3()

Examples

```
if (interactive()) {
  myproxy <- cms_zarr_proxy(
    product       = "GLOBAL_ANALYSISFORECAST_PHY_001_024",
    layer         = "cmems_mod_glo_phy-cur_anfc_0.083deg_P1D-m",
    variable      = c("uo", "vo"),
    asset         = "timeChunked")
  plot(myproxy["uo",1:200,1:100,50,1], axes = TRUE)
}
```

Index

∗ **authentication**
  cms_get_username, 8
  cms_login, 9
∗ **download**
  cms_download_native, 3
  cms_download_subset, 5
  cms_native_proxy, 10
  cms_native_s3, 12
  cms_zarr_proxy, 22
∗ **leaflet**
  cms_wmts_details, 19
∗ **product**
  cms_cite_product, 2
  cms_download_native, 3
  cms_product_details, 15
  cms_product_metadata, 16
  cms_product_services, 17
  cms_products_list, 14
∗ **supporting**
  cms_get_client_info, 7
  cms_glossary, 9
  cms_translate, 18
  cms_write_ncdf, 21

addCmsWMTSTiles *(*cms_wmts_details*)*, 19
agrepl(), *9*

cms_cite_product, 2
cms_cite_product(), *4*, *14–17*
cms_download_native, 3
cms_download_native(), *3*, *6*, *11*, *13–17*, *23*
cms_download_subset, 5
cms_download_subset(), *4*, *11*, *13*, *21*, *23*
cms_get_client_info, 7
cms_get_client_info(), *9*, *18*, *22*
cms_get_password *(*cms_get_username*)*, 8
cms_get_username, 8
cms_get_username(), *10*
cms_glossary, 9
cms_glossary(), *7*, *18*, *22*
cms_list_native_files
  *(*cms_download_native*)*, 3
cms_list_native_files(), *13*
cms_login, 9
cms_login(), *8*
cms_native_proxy, 10
cms_native_proxy(), *4*, *6*, *13*, *23*
cms_native_s3, 12
cms_native_s3(), *4*, *6*, *11*, *23*
cms_product_details, *5*, 15, *20*, *23*
cms_product_details(), *3, 4*, *14*, *16, 17*
cms_product_metadata, 16
cms_product_metadata(), *3, 4*, *14, 15*, *17*
cms_product_services, *4, 5*, *11*, 17, *20*, *23*
cms_product_services(), *3, 4*, *14–16*
cms_products_list, *2*, *4, 5*, *11*, 14, *15–17*, *20*, *23*
cms_products_list(), *3, 4*, *15–17*
cms_products_list2 *(*cms_products_list*)*, 14
cms_set_password *(*cms_get_username*)*, 8
cms_set_username *(*cms_get_username*)*, 8
cms_translate, 18
cms_translate(), *7*, *9*, *22*
cms_wmts_details, 19
cms_wmts_get_capabilities
  *(*cms_wmts_details*)*, 19
cms_write_ncdf, 21
cms_write_ncdf(), *6, 7*, *9*, *18*
cms_zarr_proxy, 22
cms_zarr_proxy(), *4*, *6*, *11*, *13*

leaflet::addWMSTiles(), *20*
leaflet::leaflet(), *20*

paws.storage::s3(), *12, 13*

stars::read_mdim(), *21*
stars::st_as_stars(), *6*, *10*, *22*

xml2::xml_new_document(), *20*