

Package ‘CopernicusMarine’

February 12, 2026

Type Package

Title Search Download and Handle Data from Copernicus Marine Service Information

Version 0.4.0

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 aws.s3, cli, dplyr, httr2 (>= 1.1.0), leaflet, lubridate, purrr, rlang, sf (>= 1.0-24), stars (>= 0.6-8), stringr, tibble, tidyr, xml2

Suggests clipr, cubelyr, knitr, lifecycle, ncmeta, rmarkdown, testthat (>= 3.0.0)

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

RoxygenNote 7.3.2

Config/testthat/edition 3

Collate 'CopernicusMarine-package.r' 'cms_catalogue_entry.R'
'cms_cite_product.r' 'generics.r' 'cms_download_native.R'
'cms_login.r' 'cms_download_subset.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
NeedsCompilation no
Repository CRAN
Date/Publication 2026-02-12 11:50:02 UTC

Contents

cms_cite_product	2
cms_download_native	3
cms_download_subset	5
cms_get_client_info	6
cms_get_username	7
cms_login	8
cms_native_proxy	9
cms_products_list	10
cms_product_details	11
cms_product_metadata	11
cms_product_services	12
cms_translate	13
cms_wmts_details	14
cms_zarr_proxy	16
Index	17

cms_cite_product	<i>How to cite a Copernicus marine product</i>
------------------	--

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 AlsoOther product-functions: [cms_product_details\(\)](#), [cms_product_services\(\)](#), [cms_products_list\(\)](#)**Examples**

```
cms_cite_product("SST_MED_PHY_SUBSKIN_L4_NRT_010_036")
```

cms_download_native	<i>Download raw files as provided to Copernicus Marine</i>
---------------------	--

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,
  ...
)

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
max	A maximum number of records to be returned.

Value

Returns NULL invisibly.

Author(s)

Pepijn de Vries

Examples

```
if (interactive()) {
  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

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

Usage

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

Arguments

username	Your Copernicus marine user name. Can be provided with <code>cms_get_username()</code> (default), or as argument here.
password	Your Copernicus marine password. Can be provided as <code>cms_get_password()</code> (default), or as argument here.
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 .
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.
timerange	A vector with two elements (lower and upper value) for a requested time range. The vector should be coercible to POSIXct.
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.
progress	A logical value. When TRUE (default) progress is reported to the console. Otherwise, this function will silently proceed.

crop	[Deprecated] . This version now uses the GDAL library to handle the subsetting and downloading of subsets. The crop argument is therefore no longer supported.
asset	Type of asset to be used when subsetting data. Should be one of "default", "ARCO", "static", "omi", or "downsampled4". 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).

Details

Currently, credentials are ignored. The subsetting service seems to be public. You can use this function without using your account. This might change in the future.

Value

Returns a `stars::st_as_stars()` object.

Author(s)

Pepijn de Vries

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

Examples

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

cms_get_username	<i>Set or get Copernicus account details</i>
------------------	--

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

Examples

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

 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 <code>cms_get_username()</code> (default), or as argument here.
password	Your Copernicus marine password. Can be provided as <code>cms_get_password()</code> (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

Examples

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

cms_native_proxy	<i>Get a proxy stars object from a native service</i>
------------------	---

Description

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, ...)
```

Arguments

product	An identifier (type character) of the desired Copernicus marine product. Can be obtained with <code>cms_products_list</code> .
layer	The name of a desired layer within a product (type character). Can be obtained with <code>cms_product_services</code> (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

Value

A **stars_proxy object**

Author(s)

Pepijn de Vries

Examples

```
if (interactive()) {
  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_products_list	<i>List products available from data.marine.copernicus.eu</i>
-------------------	---

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

<code>...</code>	Allows you to pass (search) query parameters to apply to the list. When omitted, the full list of products is returned.
<code>info_type</code>	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).

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-functions: [cms_cite_product\(\)](#), [cms_product_details\(\)](#), [cms_product_services\(\)](#)

Examples

```

cms_products_list()

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

```

cms_product_details	<i>Obtain details for a specific Copernicus marine product</i>
---------------------	--

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

Value

Returns a named list with product details.

Author(s)

Pepijn de Vries

See Also

Other product-functions: [cms_cite_product\(\)](#), [cms_product_services\(\)](#), [cms_products_list\(\)](#)

Examples

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

cms_product_metadata	<i>Obtain product meta data</i>
----------------------	---------------------------------

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

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

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.

Value

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

Author(s)

Pepijn de Vries

See Also

Other product-functions: [cms_cite_product\(\)](#), [cms_product_details\(\)](#), [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

[Experimental] 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()`

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	<i>Obtain a WMTS entry for specific Copernicus marine products and add to a leaflet map</i>
------------------	---

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",
  options = leaflet::WMSTileOptions(format = "image/png", transparent = TRUE),
  ...
)

cms_wmts_get_capabilities(product, layer, variable, type = c("list", "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 .
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 .
options	Passed on to leaflet::addWMSTiles() .
...	Passed on to leaflet::addWMSTiles() .
type	A character string indicating whether the capabilities should be returned as "list" (default) or "xml" (xml2::xml_new_document()).

Value

`cms_wmts_details` returns a tibble with details on the WMTS service. `cms_wmts_getcapabilities` returns either a list 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_zarr_proxy

*Get a proxy stars object from a Zarr service***Description**

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)
```

Arguments

product	An identifier (type character) of the desired Copernicus marine product. Can be obtained with <code>cms_products_list</code> .
layer	The name of a desired layer within a product (type character). Can be obtained with <code>cms_product_services</code> (listed as id column).
variable	The name of a desired variable in a specific layer of a product (type character). Can be obtained with <code>cms_product_details</code> .
asset	An asset that is available for the product. Should be one of "native", "wmts", "timeChunked", "downsampled4", or "geoChunked".

Value

A **stars_proxy object**

Author(s)

Pepijn de Vries

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

* product-functions

- cms_cite_product, [2](#)
- cms_product_details, [11](#)
- cms_product_services, [12](#)
- cms_products_list, [10](#)

addCmsWMTSTiles (cms_wmts_details), [14](#)

cms_cite_product, [2](#), [10–12](#)
cms_download_native, [3](#)
cms_download_subset, [5](#)
cms_get_client_info, [6](#)
cms_get_password (cms_get_username), [7](#)
cms_get_username, [7](#)
cms_list_native_files
 (cms_download_native), [3](#)
cms_login, [8](#)
cms_native_proxy, [9](#)
cms_product_details, [3](#), [5](#), [10](#), [11](#), [12](#), [15](#), [16](#)
cms_product_metadata, [11](#)
cms_product_services, [3](#), [5](#), [9–11](#), [12](#), [15](#), [16](#)
cms_products_list, [2](#), [3](#), [5](#), [9](#), [10](#), [11](#), [12](#), [15](#),
 [16](#)
cms_products_list2 (cms_products_list),
 [10](#)
cms_set_password (cms_get_username), [7](#)
cms_set_username (cms_get_username), [7](#)
cms_translate, [13](#)
cms_wmts_details, [14](#)
cms_wmts_get_capabilities
 (cms_wmts_details), [14](#)
cms_zarr_proxy, [16](#)

leaflet::addWMSTiles(), [15](#)
leaflet::leaflet(), [15](#)

stars::st_as_stars(), [6](#), [9](#), [16](#)

xml2::xml_new_document(), [15](#)