

Package ‘ncdfCF’

June 10, 2024

Type Package

Title Easy Access to NetCDF Files with CF Metadata Conventions

Version 0.1.1

Description Network Common Data Form (NetCDF) files are widely used for scientific data. Library-level access in R is provided through packages 'RNetCDF' and 'ncdf4'. Package 'ncdfCF' is built on top of 'RNetCDF' and makes the data and its attributes available as a set of S4 classes that are informed by the Climate and Forecasting Metadata Conventions. Access to the data uses standard R subsetting operators and common function forms.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Imports Cftime, methods, RNetCDF, stringr

Collate 'ncdfCF-package.R' 'ncdfResource.R' 'ncdfObject.R'
'ncdfDimension.R' 'ncdfVariable.R' 'ncdfDataset.R'
'ncdfDimensionCharacter.R' 'ncdfDimensionNumeric.R'
'ncdfDimensionTime.R' 'utils.R' 'zzz.R'

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Patrick Van Laake [aut, cre, cph]

Maintainer Patrick Van Laake <patrick@vanlaake.net>

Repository CRAN

Date/Publication 2024-06-10 17:20:02 UTC

Contents

attribute,ncdfObject-method	2
axis,ncdfDimension-method	3
dimlength	4
has_bounds,ncdfDimensionNumeric-method	4

has_bounds,ncdfDimensionTime-method	5
id,ncdfObject-method	6
indexOf	6
length,ncdfDimension-method	7
name,ncdfObject-method	8
names,ncdfDataset-method	9
ncdfDataset-class	9
ncdfDimension-class	10
ncdfDimensionCharacter-class	10
ncdfDimensionGenerics	11
ncdfDimensionNumeric-class	11
ncdfDimensionTime-class	12
ncdfDimnames	12
ncdfGenerics	13
ncdfObject-class	14
ncdfResource-class	14
ncdfVariable-class	15
open_ncdf	15
showObject	16
show_attributes,ncdfObject-method	17
subset,ncdfVariable-method	18
time,ncdfDimension-method	20
time,ncdfDimensionTime-method	20
[,ncdfVariable-method	21
[[,ncdfDataset-method	22

Index 24

attribute,ncdfObject-method
Get an attribute value

Description

Extract the value of a named attribute of the ncdfCF object. When found, the value will have the type of the attribute and it may be of type list if the attribute has multiple values.

Usage

```
## S4 method for signature 'ncdfObject'
attribute(object, att)
```

Arguments

object	A ncdfObject instance.
att	Attribute to find in \$name column.

Value

Value of the \$value column or character(0) when not found.

Examples

```
fn <- system.file("extdata",
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
                  package = "ncdfCF")
ds <- open_ncdf(fn)
lon <- ds[["lon"]]
attribute(lon, "standard_name")
```

axis,ncdfDimension-method

Dimension axis

Description

Dimension axis

Usage

```
## S4 method for signature 'ncdfDimension'
axis(x)
```

Arguments

x The ncdfDimension to get the axis for.

Value

One of X, Y, Z, T or NA_character_ when not known.

Examples

```
fn <- system.file("extdata",
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
                  package = "ncdfCF")
ds <- open_ncdf(fn)
time <- ds[["time"]]
axis(time)
```

dimlength	<i>Lengths of dimensions of the data set or variable</i>
-----------	--

Description

With this method the lengths of all dimensions of a dataset or a variable are returned. Note that a dataset may have more dimensions than a variable from the same dataset: other variables may use different dimensions.

Usage

```
## S4 method for signature 'ncdfVariable'
dim(x)

## S4 method for signature 'ncdfDataset'
dim(x)
```

Arguments

x An instance of ncdfDataset or ncdfVariable.

Value

Named integer vector of dimension sizes of dimensions in x.

Examples

```
fn <- system.file("extdata",
  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
  package = "ncdfCF")
ds <- open_ncdf(fn)

# ncdfDataset
dim(ds)

# ncdfVariable
dim(ds[["pr"]])
```

has_bounds,ncdfDimensionNumeric-method	<i>Does the dimension have 'bounds' set?</i>
--	--

Description

Does the dimension have 'bounds' set?

Usage

```
## S4 method for signature 'ncdfDimensionNumeric'  
has_bounds(x)
```

Arguments

x The ncdfDimension object to query.

Value

Logical to flag if bounds have been set or not.

Examples

```
fn <- system.file("extdata",  
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",  
                  package = "ncdfCF")  
ds <- open_ncdf(fn)  
lon <- ds[["lon"]]  
has_bounds(lon)
```

has_bounds,ncdfDimensionTime-method

Does the "time" dimension have 'bounds' set?

Description

Does the "time" dimension have 'bounds' set?

Usage

```
## S4 method for signature 'ncdfDimensionTime'  
has_bounds(x)
```

Arguments

x The ncdfDimensionTime object to query.

Value

Logical to flag if bounds have been set or not.

Examples

```
fn <- system.file("extdata",  
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",  
                  package = "ncdfCF")  
ds <- open_ncdf(fn)  
time <- ds[["time"]]  
has_bounds(time)
```

id,ncdfObject-method *Retrieve the id of an ncdfCF object*

Description

Retrieve the id of an ncdfCF object

Usage

```
## S4 method for signature 'ncdfObject'  
id(object)
```

Arguments

object The object whose id to retrieve.

Value

The integer id of the object.

Examples

```
fn <- system.file("extdata",  
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",  
                  package = "ncdfCF")  
ds <- open_ncdf(fn)  
id(ds[["lon"]])
```

indexOf

Find indices in the dimension domain

Description

Given a vector of numerical, timestamp or categorical values x , find their indices in the values of dimension y . With `method = "constant"` this returns the index of the value lower than the supplied values in x . With `method = "linear"` the return value includes any fractional part.

If bounds are set on the numerical or time dimension, the indices are taken from those bounds. Returned indices may fall in between bounds if the latter are not contiguous, with the exception of the extreme values in x .

Usage

```
## S4 method for signature 'character,ncdfDimensionCharacter'
indexOf(x, y, method = "constant")

## S4 method for signature 'numeric,ncdfDimensionNumeric'
indexOf(x, y, method = "constant")

## S4 method for signature 'ANY,ncdfDimensionTime'
indexOf(x, y, method = "constant")
```

Arguments

x	Vector of numeric, timestamp or categorical values to find dimension indices for. The timestamps can be either character, POSIXct or Date vectors. The type of the vector has to correspond to the type of y.
y	An instance of ncdfDimensionNumeric, ncdfDimensionTime or ncdfDimensionCharacter.
method	Single value of "constant" or "linear".

Value

Numeric vector of the same length as x. If method = "constant", return the index value for each match. If method = "linear", return the index value with any fractional value. Values of x outside of the range of the values in y are returned as 0 and .Machine\$integer.max, respectively.

Examples

```
fn <- system.file("extdata",
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
                  package = "ncdfCF")
ds <- open_ncdf(fn)
lon <- ds[["lon"]]
indexOf(c(8.5, 8.9, 9.3, 9.7, 10.1), lon)
indexOf(c(8.5, 8.9, 9.3, 9.7, 10.1), lon, "linear")

time <- ds[["time"]]
indexOf(c("2024-03-01", "2024-03-02"), time)
```

length,ncdfDimension-method
Length of the dimension

Description

Length of the dimension

Usage

```
## S4 method for signature 'ncdfDimension'
length(x)
```

Arguments

x The ncdfDimension to query.

Value

Integer scalar of the length of the dimension.

Examples

```
fn <- system.file("extdata",
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
                  package = "ncdfCF")
ds <- open_ncdf(fn)
time <- ds[["time"]]
length(time)
```

name,ncdfObject-method

Retrieve the name of an ncdfCF object

Description

Retrieve the name of an ncdfCF object

Usage

```
## S4 method for signature 'ncdfObject'
name(object)
```

Arguments

object The object whose name to retrieve.

Value

A character string of the name of the object.

Examples

```
fn <- system.file("extdata",
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
                  package = "ncdfCF")
ds <- open_ncdf(fn)
name(ds[["lon"]])
```

names,ncdfDataset-method
Variable names of an ncdfDataset instance

Description

Variable names of an ncdfDataset instance

Usage

```
## S4 method for signature 'ncdfDataset'
names(x)
```

Arguments

x ncdfDataset whose variable names to retrieve.

Value

A character vector of variable names.

Examples

```
fn <- system.file("extdata",
  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
  package = "ncdfCF")
ds <- open_ncdf(fn)
names(ds)
```

ncdfDataset-class *ncdfDataset class*

Description

This class represents a single NetCDF resource.

Slots

resource The ncdfResource instance that handles the NetCDF file.

keep_open Logical flag to indicate if the resource should remain open for data access after the initial read of metadata.

vars A list with the variables in the resource.

dims A list holding all the dimension data.

format A character string with the format of the NetCDF resource.

has_error logical. Flag to indicate if there was an error opening the resource.

ncdfDimension-class *Base dimension object*

Description

The 'dimension' is one of the key building blocks of a data set in an netCDF resource. This virtual class contains the common functionality of all dimension classes. Instantiable sub-classes include `ncdfDimensionNumeric` for numeric dimension values, and `ncdfDimensionTime` for "time" dimensions.

Slots

`var_id` The ID of the dimension variable if read from file.

`var_type` The type of the dimension variable if read from file.

`length` The number of elements in the dimension.

`unlim` Flag to indicate whether this dimension is unlimited, e.g. can be extended beyond the current 'length'.

`axis` The axis of the dimension, if known.

ncdfDimensionCharacter-class
Character dimension class

Description

This class describes character dimensions in an NetCDF resource. Character dimensions are discrete by default.

Slots

`values` The values of the positions along the dimension.

ncdfDimensionGenerics *Generics for ncdfCF dimensions*

Description

These are generic method definitions with implementations in descendant classes. See `ncdfDimensionNumeric`, `ncdfDimensionCharacter` and `ncdfDimensionTime` help topics for details.

Usage

`has_bounds(x)`

`axis(x)`

Arguments

`x` The ncdfCF dimension that the method operates on.

Value

`has_bounds()` returns a logical to indicate if bounds have been set on the dimension. `axis()` returns a character to indicate which of axes "X", "Y", "Z", or "T" is associated with the dimension, if any.

`ncdfDimensionNumeric-class`

Numeric dimension class

Description

This class describes numeric dimensions in an NetCDF resource.

Slots

`values` The values of the positions along the dimension.

`bounds` The bounds of the dimension values, if any.

ncdfDimensionTime-class
Dimension object

Description

The 'dimension' is one of the key building blocks of a data set in an netCDF resource. This class is specifically for dimensions that represent the "time" dimension.

Slots

values The values of the positions along the dimension, as an instance of S4 class 'CFtime'.

ncdfDimnames *Dimnames of an ncdfObject instance*

Description

Retrieve the dimension names of an ncdfCF object. The return value differs depending on the type of object:

- ncdfDataset, ncdfVariable: The dimnames are returned as a vector of the names of the dimensions of the dataset or variable. Note that this differs markedly from the base: :dimnames() functionality.
- ncdfDimensionNumeric: The values of the elements along the dimension as a numeric vector.
- ncdfDimensionCharacter: The values of the elements along the dimension as a character vector.
- ncdfDimensionTime: The values of the elements along the dimension as a character vector containing timestamps in ISO8601 format. This could be dates or date-times if time information is available in the dimension.

Usage

```
## S4 method for signature 'ncdfVariable'
dimnames(x)

## S4 method for signature 'ncdfDataset'
dimnames(x)

## S4 method for signature 'ncdfDimensionCharacter'
dimnames(x)

## S4 method for signature 'ncdfDimensionNumeric'
dimnames(x)

## S4 method for signature 'ncdfDimensionTime'
dimnames(x)
```

Arguments

x An ncdfObject whose dimension names to retrieve. This could be ncdfDataset, ncdfVariable, ncdfDimensionNumeric or ncdfDimensionTime.

Value

A vector as described in the Description section.

Examples

```
fn <- system.file("extdata",
  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
  package = "ncdfCF")
ds <- open_ncdf(fn)

# ncdfDataset
dimnames(ds)

# ncdfVariable
pr <- ds[["pr"]]
dimnames(pr)

# ncdfDimensionNumeric
lon <- ds[["lon"]]
dimnames(lon)

# ncdfDimensionTime
t <- ds[["time"]]
dimnames(t)
```

ncdfGenerics

Generics for ncdfCF objects

Description

These are generic method definitions with implementations in descendant classes. Please see the help topics for the descendant classes.

Usage

shard(object)

brief(object)

id(object)

name(object)

```
attribute(object, att)
```

```
show_attributes(object)
```

Arguments

object	The ncdfCF object that the method operates on. This includes datasets, variables, dimensions, and possible others including instances of descendant classes.
att	Name of the attribute to look up.

Value

Various. Please see the documentation of the methods in descendant classes.

ncdfObject-class	<i>Ancestor of all netCDF objects.</i>
------------------	--

Description

This is a virtual class that is the ancestor of other ncdfCF classes. Slots 'id' and 'name' are populated in the descending classes.

Slots

id	Identifier of the netCDF object.
name	Name of the netCDF object.
attributes	A data.frame holding all attributes of the object.
resource	ncdfResource instance to access the netCDF resource.

ncdfResource-class	<i>Low-level access to the RNetCDF package</i>
--------------------	--

Description

This class, of which there will be 1 instance in the package, shared by all ncdfObjects, provides low-level access to the RNetCDF package, mainly to provide a valid handle with which to access the underlying netCDF resource.

Slots

uri	character.
handle	ANY.

ncdfVariable-class *The ncdfVariable class*

Description

The ncdfVariable class

Slots

dims ncdfDimension. Vector of dimensions of this variable.

open_ncdf *Read a NetCDF resource*

Description

Read a NetCDF resource

Usage

```
open_ncdf(resource, keep_open = FALSE)
```

Arguments

resource	The name of the NetCDF resource to open, either a local file name or a remote URI.
keep_open	Logical flag to indicate if the NetCDF resource has to remain open after reading the metadata. This should be enabled typically only for programmatic access or when a remote resource has an expensive access protocol (i.e. 2FA). The resource has to be explicitly closed with <code>close()</code> after use. Note that when a dataset is opened with <code>keep_open = TRUE</code> the resource may still be closed by the operating system or the remote server.

Value

An ncdfDataset instance, or an error if the resource was not found or errored upon reading.

Examples

```
fn <- system.file("extdata",
  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
  package = "ncdfCF")
ds <- open_ncdf(fn)
ds
```

`showObject`*Summary of object details*

Description

These methods provide information on the various `ncdfCF` objects. While the individual methods are generally behaving the same for all descendant classes, there are some differences related to the nature of the objects.

- Method `show()` will provide many details of the object over multiple lines printed to the console. This includes all attributes so it could be a substantive amount of information.
- Method `brief()` returns some details of the object in a 1-row `data.frame` for further processing, such as combining details from all variables of a dataset into a single table.
- Method `shard()` returns a very short character string with some identifying properties of the object, typically only useful when combined with shards of other object to provide a succinct overview of the dataset. This method has limited usability for the user but may be of interest for programmatic access.

Usage

```
## S4 method for signature 'ncdfDimension'  
shard(object)
```

```
## S4 method for signature 'ncdfVariable'  
show(object)
```

```
## S4 method for signature 'ncdfVariable'  
brief(object)
```

```
## S4 method for signature 'ncdfVariable'  
shard(object)
```

```
## S4 method for signature 'ncdfDataset'  
show(object)
```

```
## S4 method for signature 'ncdfDataset'  
brief(object)
```

```
## S4 method for signature 'ncdfDimensionCharacter'  
show(object)
```

```
## S4 method for signature 'ncdfDimensionCharacter'  
brief(object)
```

```
## S4 method for signature 'ncdfDimensionNumeric'  
show(object)
```



```
## S4 method for signature 'ncdfDimensionNumeric'  
brief(object)  
  
## S4 method for signature 'ncdfDimensionTime'  
show(object)  
  
## S4 method for signature 'ncdfDimensionTime'  
brief(object)
```

Arguments

object The ncdfObject to show.

Value

show() prints information to the console. brief() returns a 1-row data.frame with some details of object. shard() returns a character string with a few identifying details of object.

Examples

```
fn <- system.file("extdata", "ERA5land_Rwanda_20160101.nc", package = "ncdfCF")  
ds <- open_ncdf(fn)  
  
# ncdfDataset, show  
ds  
  
# ncdfDataset, brief  
# Note that the variables and dimensions are described by shards  
brief(ds)  
  
# ncdfVariable, show  
pev <- ds[["pev"]]  
pev  
  
# ncdfDimensionNumeric, shard  
lon <- ds[["longitude"]]  
shard(lon)
```

show_attributes,ncdfObject-method

Print the attributes of the object to the console

Description

Print the attributes of the object to the console

Usage

```
## S4 method for signature 'ncdfObject'
show_attributes(object)
```

Arguments

object ncdfObject whose attributes to print.

Value

Nothing.

Examples

```
fn <- system.file("extdata",
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
                  package = "ncdfCF")
ds <- open_ncdf(fn)
show_attributes(ds[["pr"]])
```

subset,ncdfVariable-method

Extract a subset of values from a variable

Description

This method extracts a subset of values from the array of the variable, with the range along each dimension to extract expressed in values of the domain of each dimension.

Usage

```
## S4 method for signature 'ncdfVariable'
subset(x, subset, rightmost.closed = FALSE, ...)
```

Arguments

x The ncdfVariable from which to extract a subset of values.

subset A list with the range to extract from each dimension of x. The list should have elements for the dimensions to extract a subset from - if a dimension is not present in the list the entire dimension will be extracted from the array. List element names should be the axis designator X, Y, Z or T, or the name of the dimension - dimensions without a recognized axis and any additional dimensions beyond the four standard ones can only be specified by name. Axis values and dimension names are case-sensitive and can be specified in any order. If values for the range per dimension fall outside of the extent of the dimension, the range is clipped to the extent of the dimension.

```

rightmost.closed
    Single logical value to indicate if the upper boundary of range in each the di-
    mension should be included.
...
    Ignored.

```

Details

The range of values along each dimension to be subset is expressed in values of the domain of the dimension. Any dimensions for which no information is provided in the subset argument are extracted in whole. Values can be specified in a variety of ways that are specific to the nature of the dimension. For numeric dimensions it should (resolve to) be a vector of real values. A range (e.g. 100:200), a long vector (c(23, 46, 3, 45, 17), a sequence (seq(from = 78, to = 100, by = 2), all work. Note, however, that only a single range is generated from the vector so these examples resolve to 100:200, 3:46, and 78:100, respectively. For time dimensions a vector of character timestamps, POSIXct or Date values must be specified. As with numeric values, only the two extreme values in the vector will be used.

If the range of values for a dimension in subset extend the valid range of the dimension in x, the extracted slab will start at the beginning for smaller values and extend to the end for larger values. If the values envelope the valid range the entire dimension will be extracted in the result. If the range of subset values for any dimension are all either smaller or larger than the valid range of the dimension in x then nothing is extracted and NULL is returned.

As an example, to extract values of a variable for Australia for the year 2020, where the first dimension in x is the longitude, the second dimension is the latitude, both in degrees, and the third (and final) dimension is time, the values are extracted by subset(x, list(X = c(112, 154), Y = c(-9, -44), T = c("2020-01-01", "2021-01-01"))). You could take the longitude-latitude values from sf::st_bbox() or terra::ext() if you have specific spatial geometries for whom you want to extract data. Note that this works equally well for projected coordinate reference systems - the key is that the specification in argument subset uses the same domain of values as the respective dimensions in x use.

Value

An array with dimnames and other attributes set, or NULL.

Examples

```

fn <- system.file("extdata",
  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
  package = "ncdfCF")
ds <- open_ncdf(fn)
pr <- ds[["pr"]]

# Precipitation data for March for a small area
x <- subset(pr, subset = list(X = c(9, 11),
  Y = 42:45,
  T = c("2024-03-01", "2024-04-01")))

dim(x)
dimnames(x)

```

time,ncdfDimension-method

Get the full time specification of the dimension

Description

This method returns NULL. Class `ncdfDimensionTime` implements this method more usefully.

Usage

```
## S4 method for signature 'ncdfDimension'  
time(x)
```

Arguments

x A `ncdfDimension` instance.

Value

NULL

Examples

```
fn <- system.file("extdata",  
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",  
                  package = "ncdfCF")  
ds <- open_ncdf(fn)  
time(ds[["lon"]])
```

time,ncdfDimensionTime-method

Get the full time specification of the dimension

Description

This method returns the `CFtime` instance that manages all the conversions and processing for this dimension. See package `CFtime` for more details.

Usage

```
## S4 method for signature 'ncdfDimensionTime'  
time(x)
```

Arguments

x A `ncdfDimensionTime` instance.

Value

An instance of the Cftime class.

Examples

```
fn <- system.file("extdata",
                  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
                  package = "ncdfCF")
ds <- open_ncdf(fn)
time(ds[["time"]])
```

[,ncdfVariable-method *Extract data for a variable*

Description

Extract data from a `ncdfVariable` instance, optionally sub-setting the dimensions to load only data of interest.

Usage

```
## S4 method for signature 'ncdfVariable'
x[i, j, ..., drop = TRUE]
```

Arguments

<code>x</code>	An <code>ncdfVariable</code> instance to extract the data of.
<code>i, j, ...</code>	Expressions, one for each dimension of <code>x</code> , that select a number of elements along each dimension. If any expressions are missing, the entire dimension is extracted. The values for the arguments may be an integer vector or a function that returns an integer vector. The range of the values in the vector will be used. See examples, below.
<code>drop</code>	Logical, ignored. Dimensions are never dropped. Any degenerate dimensions are returned as such, with <code>dimnames</code> and appropriate attributes set.

Details

If all the data of the variable in `x` is to be extracted, simply use `[]` (unlike with regular arrays, this is required, otherwise the details of the variable are printed on the console).

The indices into the dimensions to be subset can be specified in a variety of ways; in practice it should (resolve to) be a vector of integers. A range (e.g. `100:200`), an explicit vector (`c(23, 46, 3, 45, 17)`), a sequence (`seq(from = 78, to = 100, by = 2)`), all work. Note, however, that only a single range is generated from the vector so these examples resolve to `100:200`, `3:46`, and `78:100`, respectively. It is also possible to use a custom function as an argument.

This method works with "bare" indices into the dimensions of the array. If you want to use domain values of the dimensions (e.g. longitude values or timestamps) to extract part of the variable array, use the `subset()` method.

Value

An array with dimnames and other attributes set.

Examples

```
fn <- system.file("extdata",
  "pr_day_EC-Earth3-CC_ssp245_r1i1p1f1_gr_20240101-20241231_vncdfCF.nc",
  package = "ncdfCF")
ds <- open_ncdf(fn)
pr <- ds[["pr"]]

# How are the dimensions organized?
dimnames(pr)

# Precipitation data for March for a single location
x <- pr[5, 12, 61:91]
str(x)

# Summer precipitation over the full spatial extent
summer <- pr[, , 173:263]
str(summer)
```

[[,ncdfDataset-method *Get a variable object or a dimension object from a dataset*

Description

This method can be used to retrieve a variable or a dimension from the dataset by name.

Usage

```
## S4 method for signature 'ncdfDataset'
x[[i]]
```

Arguments

x An ncdfDataset to extract a variable or a dimension from.

i The name of a variable or dimension in x.

Value

An instance of ncdfVariable or an ncdfDimension descendant class, or NULL if the name is not found.

Examples

```
fn <- system.file("extdata", "ERA5land_Rwanda_20160101.nc", package = "ncdfCF")
ds <- open_ncdf(fn)
v1 <- names(ds)[1]
var <- ds[[v1]]
var
```

Index

[,ncdfVariable-method, 21
[[,ncdfDataset-method, 22

attribute (ncdfGenerics), 13
attribute,ncdfObject-method, 2
axis (ncdfDimensionGenerics), 11
axis,ncdfDimension-method, 3

bracket_select ([,ncdfVariable-method),
21
brief (ncdfGenerics), 13
brief,ncdfDataset-method (showObject),
16
brief,ncdfDimensionCharacter-method
(showObject), 16
brief,ncdfDimensionNumeric-method
(showObject), 16
brief,ncdfDimensionTime-method
(showObject), 16
brief,ncdfVariable-method (showObject),
16

dim,ncdfDataset-method (dimlength), 4
dim,ncdfVariable-method (dimlength), 4
dimlength, 4
dimnames,ncdfDataset-method
(ncdfDimnames), 12
dimnames,ncdfDimensionCharacter-method
(ncdfDimnames), 12
dimnames,ncdfDimensionNumeric-method
(ncdfDimnames), 12
dimnames,ncdfDimensionTime-method
(ncdfDimnames), 12
dimnames,ncdfVariable-method
(ncdfDimnames), 12

has_bounds (ncdfDimensionGenerics), 11
has_bounds,ncdfDimensionNumeric-method,
4
has_bounds,ncdfDimensionTime-method, 5

id (ncdfGenerics), 13
id,ncdfObject-method, 6
indexOf, 6
indexOf,ANY,ncdfDimensionTime-method
(indexOf), 6
indexOf,character,ncdfDimensionCharacter-method
(indexOf), 6
indexOf,numeric,ncdfDimensionNumeric-method
(indexOf), 6

length,ncdfDimension-method, 7

name (ncdfGenerics), 13
name,ncdfObject-method, 8
names,ncdfDataset-method, 9
ncdfDataset-class, 9
ncdfDimension-class, 10
ncdfDimensionCharacter-class, 10
ncdfDimensionGenerics, 11
ncdfDimensionNumeric-class, 11
ncdfDimensionTime-class, 12
ncdfDimnames, 12
ncdfGenerics, 13
ncdfObject-class, 14
ncdfResource-class, 14
ncdfVariable-class, 15

open_ncdf, 15

shard (ncdfGenerics), 13
shard,ncdfDimension-method
(showObject), 16
shard,ncdfVariable-method (showObject),
16
show,ncdfDataset-method (showObject), 16
show,ncdfDimensionCharacter-method
(showObject), 16
show,ncdfDimensionNumeric-method
(showObject), 16
show,ncdfDimensionTime-method
(showObject), 16

show,ncdfVariable-method (showObject),
16
show_attributes (ncdfGenerics), 13
show_attributes,ncdfObject-method, 17
showObject, 16
subset(), 21
subset,ncdfVariable-method, 18

time,ncdfDimension-method, 20
time,ncdfDimensionTime-method, 20