

Package ‘rdwd’

June 9, 2017

Title Select and Download Climate Data from 'DWD' (German Weather Service)

Version 0.8.0

Date 2017-06-09

Depends R(>= 2.10)

Imports berryFunctions (>= 1.14.5), pbapply

Suggests RCurl, leaflet, knitr, rmarkdown, testthat, XML, data.table

Author Berry Boessenkool

Maintainer Berry Boessenkool <berry-b@gmx.de>

Description Handle climate data from the 'DWD' ('Deutscher Wetterdienst', see <https://www.dwd.de/EN/climate_environment/cdc/cdc.html> for more information). Choose files with 'selectDWD()', download and process data sets with 'dataDWD()' and 'readDWD()'.

License GPL (>= 2)

Encoding UTF-8

URL <https://github.com/brry/rdwd>

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-06-09 13:30:41 UTC

R topics documented:

createIndex	2
dataDWD	4
dirDWD	6
findID	7
index	8
indexFTP	9

lldist	11
metaInfo	12
nearbyStations	12
rdwd	14
readDWD	15
rowDisplay	16
selectDWD	17

Index 21

createIndex	<i>Create file and meta index of the DWD CDC FTP Server</i>
-------------	---

Description

This is mainly an internal function. Create data.frames out of the vector index returned by [indexFTP](#). For [fileIndex](#) (the first output element) createIndex tries to obtain res, var, per, file, id, start and end from the paths. If meta=TRUE, [metaIndex](#) and [geoIndex](#) are also created. They combine all Beschreibung files into a single data.frame.

If you create your own index as suggested in selectDWD (argument findex), you can read the produced file as shown in the example section.

Usage

```
createIndex(paths,
  base = "ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate",
  dir = "DWDdata", fname = "fileIndex.txt", meta = FALSE,
  metadir = "meta", mname = "metaIndex.txt", gname = "geoIndex.txt",
  quiet = FALSE, ...)
```

Arguments

paths	Char: vector of DWD paths returned by indexFTP called with the same base value as this function
base	Main directory of DWD ftp server, defaulting to observed climatic records. DEFAULT: ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate
dir	Char: writeable directory name where to save the main output(s). Created if not existent. DEFAULT: "DWDdata" at current getwd()
fname	Char: Name of file in dir in which to write fileIndex . Use fname="" to suppress writing. DEFAULT: "fileIndex.txt"
meta	Logical: should metaIndex also be created from fileIndex? Uses dataDWD to download files if not present. DEFAULT: FALSE
metadir	Char: Directory (subfolder of dir) where original description files are downloaded to if meta=TRUE. vPassed to dataDWD . "" to write in dir. DEFAULT: "meta"
mname	Char: Name of file in dir (not metadir) in which to write metaIndex . Use mname="" to suppress writing. DEFAULT: "metaIndex.txt"

gname	Filename for geoIndex . DEFAULT: "geoIndex.txt"
quiet	Logical: Suppress messages about progress and filenames? DEFAULT: FALSE
...	Further arguments passed to dataDWD for the meta part.

Value

invisible data.frame (or if meta=TRUE, list with two data.frames) with a number of columns inferred from the paths. Each is also written to disc.

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Oct-Nov 2016, June 2017

See Also

[indexFTP](#), [fileIndex](#), [metaIndex](#), [selectDWD](#)

Examples

```
## Not run: # Not tested with R CMD check because of file writing
link <- "/daily/kl/historical/tageswerte_00699_19490101_19580630_hist.zip"
ind <- createIndex(link, dir=tempdir())
ind
link2 <- "/daily/kl/historical/KL_Tageswerte_Beschreibung_Stationen.txt"
link3 <- "/daily/kl/recent/KL_Tageswerte_Beschreibung_Stationen.txt"
ind2 <- createIndex(c(link,link2,link3), dir=tempdir(), meta=TRUE)
lapply(ind2, head)

## End(Not run)

# For real usage, see last part of
if(interactive())
browseURL("https://github.com/brry/rdwd/blob/master/R/rdwd-package.R")
# where the Indexes are added to the package

# Read results in later:
## Not run: ## files normally not yet available:
fileIndex2 <- read.table("DWDdata/fileIndex.txt", sep="\t", header=TRUE,
                        colClasses="character")
metaIndex2 <- read.table("DWDdata/metaIndex.txt", sep="\t", header=TRUE, as.is=TRUE)

## End(Not run)
```

dataDWD

*Download data from the DWD CDC FTP Server***Description**

Get climate data from the German Weather Service (DWD) FTP-server. The desired .zip (or .txt) dataset is downloaded into dir. If read=TRUE, it is also read, processed and returned as a data.frame.

Usage

```
dataDWD(file, dir = "DWDdata", force = FALSE, sleep = 0, quiet = FALSE,
        progbar = !quiet, browse = FALSE, read = TRUE, meta = substr(file,
        nchar(file) - 3, 10000) == ".txt", format = NA, ntrunc = 2, ...)
```

Arguments

file	Char (vector): complete file URL(s) (including base and filename.zip) as returned by selectDWD . Can be a vector with several filenames.
dir	Char: Writeable directory name where to save the downloaded file. Created if not existent. DEFAULT: "DWDdata" at current getwd()
force	Logical: always download, even if the file already exists in dir? If FALSE, it is still read (or name returned). DEFAULT: FALSE
sleep	Number. If not 0, a random number of seconds between 0 and sleep is passed to Sys.sleep after each download to avoid getting kicked off the FTP-Server. DEFAULT: 0
quiet	Logical: suppress message about directory / filenames? DEFAULT: FALSE
progbar	Logical: present a progress bar with estimated remaining time? If missing and length(file)==1, progbar is internally set to FALSE. Only works if the R package pbapply is available. DEFAULT: TRUE (!quiet)
browse	Logical: open repository via browseURL and return URL folder path? If TRUE, no data is downloaded. If file has several values, only unique folders will be opened. DEFAULT: FALSE
read	Logical: read the file(s) with readDWD ? If FALSE, only download is performed and the filename(s) returned. DEFAULT: TRUE
meta	Logical (vector): is the file a meta file? Passed to readDWD . DEFAULT: TRUE for each file ending in ".txt"
format	Char (vector): format used in strptime to convert date/time column, see readDWD . DEFAULT: NA
ntrunc	Single integer: number of filenames printed in messages before they get truncated with message "(and xx more)". DEFAULT: 2
...	Further arguments passed to download.file

Value

Presuming downloading and processing were successful: if `read=TRUE`, a data.frame of the desired dataset (as returned by `readDWD`), otherwise the filename as saved on disc (may have "_n" appended in name, see `newFilename`).

If `length(file)>1`, the output is a list of data.frames / vector of filenames.

The output is always invisible.

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Jun-Oct 2016

See Also

`selectDWD`, `readDWD`, `download.file`. Helpful for plotting: `berryFunctions::monthAxis`, see also `berryFunctions::climateGraph`

Examples

```
## Not run: ## requires internet connection
# find files for a given station name and file path:
link <- selectDWD("Kupferzell-Rechbach", res="hourly", var="precipitation", per="recent")
# actually download and read files
prec <- dataDWD(link, dir="DWDdata") # the default dir
fname <- dataDWD(link, read=FALSE) # filename, no second download (unless force=TRUE)

# current and historical files:
link <- selectDWD("Potsdam", res="daily", var="kl", per="hr", outvec=TRUE); link
potsdam <- dataDWD(link)
potsdam <- do.call(rbind, potsdam) # this will partly overlap in time
plot(LUFTTEMPERATUR~MESS_DATUM, data=tail(potsdam,1000), type="l")
# Straight line marks the jump back in time
# check for equality:
dup <- which(duplicated(potsdam$MESS_DATUM))
dup_df <- which(duplicated(potsdam))
err <- dup[ ! dup %in% dup_df]
err <- potsdam[potsdam$MESS_DATUM %in% potsdam$MESS_DATUM[err], ]
err <- err[order(err$MESS_DATUM),]
View(err) # WINDGESCHWINDIGKEIT (wind speed) has been slightly changed
# Keep only historical dataset:
potsdam <- potsdam[!duplicated(potsdam$MESS_DATUM),]

# several files:
link <- c(link, selectDWD("Potsdam", res="daily", var="kl", per="hr", outvec=TRUE))
clim <- dataDWD(link)
fname <- dataDWD(link, read=FALSE)
clim <- readDWD(fname)
unzip(zipfile=paste0("DWDdata/",fname[1]), exdir="DWDdata/Testunzip")
# There's quite some important meta information there!

plot(prec$MESS_DATUM, prec$NIEDERSCHLAGSHOEHE, main="DWD hourly rain Kupferzell", col="blue",
      xaxt="n", las=1, type="l", xlab="Date", ylab="Hourly rainfall [mm]")
```

```
monthAxis(1, ym=T)

d <- dataDWD(selectDWD(id="05692", res="daily", var="kl", per="recent"))
# writes into the same folder (dir="DWDdata")

folder <- dataDWD(link, browse=T)
folder

# With many files, use sleep
links <- selectDWD(res="daily", var="solar", meta=FALSE)
sol <- dataDWD(links, sleep=20) # random waiting time after download (0 to 20 secs)

# Real life example with data completeness check etc:
browseURL("http://github.com/brry/prectemp/blob/master/Code_example.R")

## End(Not run)
```

dirDWD

directory management for rdwd

Description

Manage directories with useful messages in the rdwd package.

Usage

```
dirDWD(dir = "DWDdata", quiet = FALSE)
```

Arguments

dir	Char for dirDWD: writeable directory name. Created if not existent. DEFAULT: "DWDdata" at current getwd()
quiet	Logical: Suppress messages about creating dir? DEFAULT: FALSE

Value

dirDWD invisibly returns the prior working directory as per [setwd](#).

Author(s)

Berry Boessenkool, <brry-b@gmx.de>, Oct 2016

See Also

[dataDWD](#)

Examples

```
# see source code of dataDWD and metaDWD

## Not run: ## folder creation + deletion
owd <- getwd(); owd
dirDWD("dummydummydummy")
getwd()
setwd(owd)
dirDWD("dummydummydummy")
setwd(owd)
unlink("dummydummydummy")

## End(Not run)
```

findID	<i>find DWD weather station ID from name</i>
--------	--

Description

Identify DWD weather station ID from station name

Usage

```
findID(name = "", exactmatch = TRUE, mindex = metaIndex)
```

Arguments

name	Char: station name(s) that will be matched in mindex to obtain id . DEFAULT: ""
exactmatch	Logical: Should name match an entry in mindex exactly (be ==)? If FALSE, name may be a part of mindex\$Stationsname, as checked with grepl . This is useful e.g. to get all stations starting with a name (e.g. 42 IDs for Berlin). DEFAULT: TRUE
mindex	Single object: Index used to select id if name is given. DEFAULT: <code>rdwd:::metaIndex</code>

Value

Character string (vector) with ID(s)

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Oct-Nov 2016

See Also

used in [selectDWD](#), [metaInfo](#)

Examples

```
## Not run:
# Give weather station name (must be existing in metaIndex):
findID("Potsdam")
findID("potsDam") # capitalization is ignored
# all names containing "Hamburg":
findID("Hamburg", exactmatch=FALSE)
findID("Potsdam", exactmatch=FALSE)

# vectorized:
findID(c("Potsdam", "Berlin-Buch"))

# German Umlauts are changed to ue, ae, oe, ss
findID("Muenchen", FALSE)
berryFunctions::convertUmlaut("M?nchen") # use this to convert umlauts in lists

# See if warnings come as expected and are informative:
findID("this_is_not_a_city")
findID(c("Wuppertal", " this_is_not_a_city") )

findID()
findID(7777)
findID("01050")

## End(Not run)
```

index

Indexes of files and metadata on the DWD CDC FTP server

Description

Created with [createIndex](#) in the last section of <https://github.com/brry/rdwd/blob/master/R/rdwd-package.R> In functions, you can access them with `rdwd::fileIndex` etc.

fileIndex: A data.frame with the filenames at the default base value ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate/.

metaIndex: A data.frame with the contents of all the station description files (..._Beschreibung_Stationen.txt) in the folders hourly, daily and monthly at base.

geoIndex: metaIndex distilled to geographic locations.

Format

fileIndex: data.frame with character strings. ca 25k rows x 7 columns:

res, var, per (see [selectDWD](#)), station id and time series start and end according to path.

metaIndex: data.frame with ca 35k rows for 12 columns:

Stations_id, von_datum, bis_datum, Stationshoehe, geoBreite, geoLaenge, Stationsname, Bundeslan

geoIndex: data.frame with ca 6k rows for 11 columns:

id, name, state, lat, lon, ele, nfiles, nonpublic, recentfile, display, col

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, June-Nov 2016, June 2017

Source

Deutscher WetterDienst / Climata Data Center FTP Server

See Also

[createIndex](#), [indexFTP](#), [selectDWD](#), [findID](#), [metaInfo](#), [vignette\("mapDWD"\)](#)

Examples

```
data(fileIndex)
data(metaIndex)
data(geoIndex)
head(fileIndex)
head(metaIndex)
head(geoIndex)

# in functions, you can use head(rdwd::fileIndex) etc, but I don't export them
# because Hadley says 'Never @export a data set' in
# browseURL("http://r-pkgs.had.co.nz/data.html#data-data")
```

indexFTP

Create a recursive index of the DWD CDC FTP Server

Description

Create a list of all the files (in subfolders) at the Climate Data Center (CDC) FTP-Server from the German Weather Service (DWD, Deutscher WetterDienst) at ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate.

The R package RCurl must be available to do this. If RCurl::getURL fails, usually because bot access is detected and denied, there will still be an output which you can pass in a second run via folder to extract the remaining dirs. You might want to wait a bit and set sleep to a higher value in that case.

Usage

```
indexFTP(folder = "currentfindex",
  base = "ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate",
  sleep = 0, dir = "DWDdata", filename = folder[1], overwrite = FALSE,
  quiet = FALSE, progbar = !quiet, verbose = FALSE)
```

Arguments

folder	Folder(s) to be indexed recursively, e.g. "/hourly/wind/". If it is "currentindex" (the default) and base is the default, it is changed to all folders in current <code>fileIndex</code> : <code>unique(dirname(fileIndex\$path))</code> . DEFAULT: "currentindex"
base	Main directory of DWD ftp server, defaulting to observed climatic records. DEFAULT: <code>ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate</code>
sleep	If not 0, a random number of seconds between 0 and <code>sleep</code> is passed to <code>Sys.sleep</code> after each read folder to avoid getting kicked off the FTP-Server. DEFAULT: 0
dir	Writeable directory name where to save the downloaded file. Created if not existent. DEFAULT: "DWDdata" at current <code>getwd()</code>
filename	Character: Part of output filename. "INDEX_of_DWD_" is prepended, "/" replaced with "_", ".txt" appended. DEFAULT: <code>folder[1]</code>
overwrite	Logical: Overwrite existing file? If not, "_n" is added to the filename, see <code>berryFunctions::newFilename</code> . DEFAULT: FALSE
quiet	Suppress progbars and message about directory/files? DEFAULT: FALSE
progbar	Logical: present a progress bar in each level? Only works if the R package <code>pbapply</code> is available. DEFAULT: TRUE
verbose	Logical: write a lot of messages from <code>RCurl::getURL</code> ? DEFAULT: FALSE (usually, you dont need all the curl information)

Details

It's not suggested to run this for all folders, as it can take quite some time and you may get kicked off the FTP-Server. This package contains an index of the climatic observations at weather stations: `View(rdwd::fileIndex)` If it is out of date, please let me know!

Value

currently a vector with file paths (output may change in the future)

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Oct 2016

Examples

```
## Not run: ## Needs internet connection
sol <- indexFTP(folder="/daily/solar")
head(sol)

mon <- indexFTP(folder="/monthly/k1", verbose=TRUE)

## End(Not run)
```

lldist	<i>distance between lat-long coordinates</i>
--------	--

Description

Great-circle distance between points at lat-long coordinates. Mostly a copy of `OSMscale::earthDist` Version 0.4.5 (2017-03-02). <https://github.com/brry/OSMscale/blob/master/R/earthDist.R#L57-L102>. Copied manually to avoid dependency hell. Does not check coordinates. Not exported.

Usage

```
lldist(lat, long, data, r = 6371, i = 1L)
maxlldist(lat, long, data, r = 6371, fun = max, each = TRUE, ...)
```

Arguments

lat, long	Latitude (North/South) and longitude (East/West) coordinates in decimal degrees
data	Optional: data.frame with the columns lat and long
r	radius of the earth. Could be given in miles. DEFAULT: 6371 (km)
i	Integer: Index element against which all coordinate pairs are computed. DEFAULT: 1
fun	Function to be applied. DEFAULT: <code>max</code>
each	Logical: give max dist to all other points for each point separately? If FALSE, will return the maximum of the complete distance matrix, as if <code>max(maxlldist(y, x))</code> . For examples, see <code>OSMscale::maxEarthDist</code> DEFAULT: TRUE
...	Further arguments passed to fun, like <code>na.rm=TRUE</code>

Value

Vector with distance(s) in km (or units of r, if r is changed)

Author(s)

Berry Boessenkool, <berryy-b@gmx.de>, Aug 2016 + Jan 2017. Angle formula from Diercke Weltatlas 1996, Page 245

`metaInfo`*Information for a station ID on the DWD CDC FTP server*

Description

Information for a station ID on the DWD CDC FTP server

Usage

```
metaInfo(id, hasfileonly = TRUE)
```

Arguments

<code>id</code>	Station ID (integer number or convertible to one)
<code>hasfileonly</code>	Logical: Only show entries that have files? DEFAULT: TRUE

Value

invisible data.frame. Also [prints](#) the output nicely formatted.

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Nov 2016

See Also

[metaIndex](#)

Examples

```
metaInfo(2849)
```

`nearbyStations`*Find DWD stations close to given coordinates*

Description

Select DWD stations within a given radius around a set of coordinates

Usage

```
nearbyStations(lat, lon, radius, res = NA, var = NA, per = NA,  
mindate = NA, hasfileonly = TRUE,  
statname = "nearbyStations target location", quiet = FALSE, ...)
```

Arguments

lat, lon	Coordinates [degrees N/S, E/W]
radius	Maximum distance [km] within which stations will be selected
res, var, per	Restrictions for dataset type as documented in selectDWD . Each can be a vector of entries. DEFAULTS: NA (ignored)
mindate	Minimum dataset ending date (as per metadata). Integer in the form of YYYYMMDD, e.g. 20170301. DEFAULT: NA
hasfileonly	Logical: only return entries for which there is an open-access file available? DEFAULT: TRUE
statname	Character: name for target locatio. DEFAULT: "nearbyStations target location"
quiet	Logical: suppress progress messages? DEFAULT: FALSE
...	Further arguments passed to selectDWD

Value

[metaIndex](#) subset with additional columns "dist" and "url"

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Mar 2017

See Also

[selectDWD](#), [metaIndex](#)

Examples

```
# 1. Basic usage ----

m <- nearbyStations(49.211784, 9.812475, radius=30,
  res=c("daily","hourly"), var= c("precipitation","more_precip","kl") ,
  mindate=20160530, statname="Braunsbach catchment center")

# 2. Remove duplicates ----

# if kl and more_precip are both available, keep only more_precip:
library("berryFunctions")
m <- sortDF(m, "var")
m <- m[!duplicated(paste0(m$Stations_id, m$res)),]
m <- sortDF(m, "res")
m <- sortDF(m, "dist", decreasing=FALSE)
rownames(m) <- NULL

## 3. Interactive map ----
```

```

## Not run: ## Excluded from CRAN checks because of leaflet dependency
library(leaflet)
m$col <- "red" ; m$col[1] <- "blue"
leaflet(m) %>% addTiles() %>%
  addCircles(lng=9.812475, lat=49.211784, radius=30e3) %>%
  addCircleMarkers(~geoLaenge, ~geoBreite, col=~col, popup=~Stationsname)

## End(Not run)

## 4. Download and process data ----

## Not run: ## Excluded from CRAN checks because of data download
# Download and process data for the stations, create a list of data.frames:
prec <- dataDWD(m$url) # once downloaded, will only read
names(prec) <- m$Stations_id[-1]
prec29 <- sapply(prec[m$res[-1]=="daily"], function(x)
  x[x$MESS_DATUM==as.POSIXct(as.Date("2016-05-29")), c("STATIONS_ID", "NIEDERSCHLAGSHOEHE")])
prec29 <- data.frame(Stations_id=unlist(prec29[1,]), precsum=unlist(prec29[2,]))
prec29 <- merge(prec29, m[m$res=="daily",c(1,4:7,14)], sort=FALSE)
View(prec29)

## End(Not run)

## 5. Plot rainfall sum on map

## Not run: ## Excluded from CRAN checks because of map download
plot(geoBreite~geoLaenge, data=m, asp=1)
textField(prec29$geoLaenge, prec29$geoBreite, prec29$precsum, col=2)

# If OSMscale installation fails, go to:
browseURL("https://github.com/brry/OSMscale#installation")
# install.packages("OSMscale")
library(OSMscale)
map <- pointsMap(geoBreite,geoLaenge, data=m, type="maptoolkit-topo")
pp <- projectPoints("geoBreite", "geoLaenge", data=prec29, to=map$stiles[[1]]$projection)
prec29 <- cbind(prec29,pp) ; rm(pp)
plot(map, removeMargin=FALSE)
scaleBar(map, cex=1.5, type="line", y=0.82)
title(main="Rainfall sum 2016-05-29 7AM-7AM [mm]", line=-1)
textField(prec29$x, prec29$y, round(prec29$precsum), font=2, cex=1.5)

## End(Not run)

```

Description

Select weather data from the DWD (Deutscher Wetterdienst) with [selectDWD](#) or [nearbyStations](#).
Download and process data sets with [dataDWD](#) and [readDWD](#).
Station selection is done offline with [fileIndex](#) and [findID](#) (which uses [metaIndex](#)).
The Index objects are created with [indexFTP](#) and [createIndex](#).
For an introduction to the package, see the [main vignette](#).

Searchability Terms

Weather Data Germany download with R, Climate Data Germany
Deutscher Wetterdienst R Daten download Klimastationen
DWD Daten mit R runterladen, Wetter und Klimadaten in R

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, June-Nov 2016, June 2017

See Also

USA data: [countyweather](#), [rnoaa](#)
World data: [Global Surface Summary of the Day](#)

readDWD

Process data from the DWD CDC FTP Server

Description

Read climate (meta) data that was downloaded with [dataDWD](#). The file is read, processed and returned as a data.frame.

file can be a vector with several filenames. The arguments meta and format can also be a vector and will be recycled to the length of file.

If meta=TRUE, column widths for [read.fwf](#) are computed internally. If needed, readDWD tries to set the locale to German (to handle Umlaute correctly). They can then be processed with `dd$Stations_id <- berryFunctions::convertUmlaut(dd$Stations_id)`.

Usage

```
readDWD(file, meta = substr(file, nchar(file) - 3, 10000) == ".txt",  
        fread = requireNamespace("data.table", quietly = TRUE), html = FALSE,  
        format = NA, tz = "GMT", progbar = TRUE)
```

Arguments

file	Char (vector): name(s) of the file(s) downloaded with <code>dataDWD</code> , e.g. "DWD-data/tageswerte_KL_02575_akt.zip" or "DWDdata/RR_Stundenwerte_Beschreibung_Stationen.txt"
meta	Logical (vector): is the file a meta file? DEFAULT: TRUE for each file ending in ".txt"
fread	Logical: read faster with <code>data.table::fread</code> ? For 30 daily/kl/hist files, 7 instead of 10 seconds. DEFAULT: TRUE if <code>data.table</code> is available.
html	Logical: read the html file instead of actual data? Requires XML to be installed. Experimental: use at own risk! DEFAULT: FALSE
format	Char (vector): if <code>meta=FALSE</code> : Format passed to <code>as.POSIXct</code> (see <code>strptime</code>) to convert the date/time column to POSIX time format. If NULL, no conversion is performed (date stays a factor). If NA, <code>readDWD</code> tries to find suitable format based on the number of characters. DEFAULT: NA
tz	Char (vector): time zone for <code>as.POSIXct</code> . "" is the current time zone, and "GMT" is UTC (Universal Time, Coordinated). DEFAULT: "GMT"
progbar	Logical: present a progress bar with estimated remaining time? If missing and <code>length(file)==1</code> , <code>progbar</code> is internally set to FALSE. Only works if the R package <code>pbapply</code> is available. DEFAULT: TRUE

Value

Invisible `data.frame` of the desired dataset, or a list of `data.frames` if `length(file) > 1`.

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Jul-Oct 2016

See Also

[dataDWD](#), [selectDWD](#)

Examples

```
# see dataDWD
```

rowDisplay

Create leaflet map popup from data.frame rows

Description

Create display character string for leaflet map popup from `data.frame` rows. This function is not exported, as it is only internally useful. A generic version is available in `berryFunctions::popleaf`.

Usage

```
rowDisplay(x)
```

Arguments

x data.frame with colnames

Value

Vector of characterstrings, one for each row in x.

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Feb 2017

See Also

[geoIndex](#)

selectDWD

Select data from the DWD CDC FTP Server

Description

Select files for downloading with [dataDWD](#). All arguments (except for `mindex`, `findex` and `base`) can be a vector and will be recycled to the maximum length of all arguments. If that length > 1, the output is a list of filenames (or vector with `outvec=TRUE`).

If station name is given, but `id` is empty (""), **id** is inferred via `mindex`. If `res/var/per` are given and valid (existing in `findex`), they are pasted together to form a **path**. Here is an overview of the behaviour in each case of availability:

case	id	path	output
1	""	""	base (and some warnings)
2	"xx"	""	All file names (across paths) for station id
3	""	"xx"	The zip file names at path
4	"xx"	"xx"	Regular single data file name

For case 3 and 4 (**path** given), you can set `meta=TRUE`. Then `selectDWD` will return the name of the station description file at **path**. This is why case 3 with `meta=FALSE` only returns the data file names (ending in `.zip`).

The following folders in `res/var/per` notation (resolution/variable/period) are available at ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate/ (and a few more at the `res` level).

"<" signifies a split into the folders `per = "recent"` and `"historical"`.

"-" signifies that there are no further sub-folders.

res=**hourly**

| res=**daily**

| res=**monthly**

```

var=
      | kl <          | kl <
      | more_precip < | more_precip <
air_temperature < |
cloudiness <      |
precipitation <   |
pressure <        |
sun <             |
wind <            |
soil_temperature < | soil_temperature < |
solar -           | solar -           |

```

Usage

```

selectDWD(name = "", exactmatch = TRUE, mindex = metaIndex,
  id = findID(name, exactmatch = exactmatch, mindex = mindex),
  base = "ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate",
  res = "", var = "", per = "", findex = fileIndex, current = FALSE,
  meta = FALSE, outvec = FALSE, ...)

```

Arguments

name	Char: station name(s) passed to <code>findID</code> , along with the next two arguments. All ignored if id is given. DEFAULT: ""
exactmatch	Logical passed to <code>findID</code> : match name with ==)? Else with <code>grepl</code> . DEFAULT: TRUE
mindex	Single object: Index with metadata passed to <code>findID</code> . DEFAULT: <code>rdwd::metaIndex</code>
id	Char/Number: station ID with or without leading zeros, e.g. "00614" or 614. Is internally converted to an integer, because some DWD meta data files also contain no leading zeros. DEFAULT: <code>findID(name)</code>
base	Single char: main directory of DWD ftp server, defaulting to observed climatic records. Must be the same base used to create <code>findex</code> . DEFAULT: <code>ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate</code>
res	Char: temporal resolution available at base, usually one of <code>c("hourly", "daily", "monthly")</code> , see section 'Description' above. <code>res/var/per</code> together form the path . DEFAULT: ""
var	Char: weather variable of interest. Usually one of <code>c("air_temperature", "cloudiness", "precipitation")</code> (only in hourly), <code>c("soil_temperature", "solar")</code> (in hourly and daily), or <code>c("kl", "more_precip")</code> (in daily and monthly). See more in <code>View(rdwd::fileIndex)</code> . DEFAULT: ""
per	Char: desired time period . One of "recent" (data from the last year, up to date usually within a few days) or "historical" (long time series). Can be abbreviated (if the first letter is "r" or "h", full names are used). To get both datasets, use <code>per="hr"</code> or <code>per="rh"</code> (and <code>outvec=TRUE</code>). <code>per</code> is set to "" if <code>var=="solar"</code> . DEFAULT: ""

findex	Single object: Index used to select filename, as returned by <code>createIndex</code> . To use a current / custom index, use <code>myIndex <- createIndex(indexFTP("/daily/solar"))</code> (with desired path, of course). DEFAULT: <code>rdwd::fileIndex</code>
current	Single logical for case 3/4 with given path: instead of <code>findex</code> , use a list of the currently available files at <code>base/res/var/per?</code> This will call <code>indexFTP</code> , thus requires availability of the <code>Rcurl</code> package. DEFAULT: <code>FALSE</code>
meta	Logical: return metadata txt file name instead of climate data zip file? Relevant only in case 4 (path and id given). See <code>metaIndex</code> for a compilation of all <code>metaData</code> files. DEFAULT: <code>FALSE</code>
outvec	Single logical: if path or ID length > 1, instead of a list, return a vector? (via <code>unlist</code>). DEFAULT: <code>FALSE</code>
...	Further arguments passed to <code>indexFTP</code> if <code>current=TRUE</code> , like <code>dir</code> , <code>quiet</code>

Value

Character string with file path and name(s) in the format "base/res/var/per/filename.zip"

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Oct 2016

See Also

`dataDWD`, `metaIndex`, [../doc/mapDWD.html](#), `vignette("mapDWD", package="rdwd")`

Examples

```
# Give weather station name (must be existing in metaIndex):
findID("Potsdam", exactmatch=FALSE)
selectDWD("Potsdam", res="daily", var="kl", per="historical")
# all files for all stations matching "Koeln":
selectDWD("Koeln", exactmatch=FALSE)
findID("Koeln", FALSE)

# or directly give station ID:
selectDWD(id="00386", res="daily", var="kl", per="historical")
selectDWD(id=386, res="daily", var="kl", per="historical")
# period abbreviatable:
selectDWD(id="00386", res="daily", var="kl", per="h")
selectDWD(id="00386", res="daily", var="kl", per="h", meta=TRUE)

# vectorizable:
selectDWD(id="01050", res="daily", var="kl", per=c("r","h"))
selectDWD(id="01050", res="daily", var="kl", per="rh", outvec=TRUE)
selectDWD(id="01050", res=c("daily","monthly"), var="kl", per="r")
# vectorization gives not the outer product, but elementwise comparison:
selectDWD(id="01050", res=c("daily","monthly"), var="kl", per="hr")

# all zip files in all paths matching id:
selectDWD(id=c(1050, 386))
```

```
# all zip files in a given path (if ID is empty):
head( selectDWD(id="", res="daily", var="kl", per="recent") )

# See if warnings come as expected and are informative:
selectDWD()
selectDWD(7777)
selectDWD(id=7777)
selectDWD(id="", res="dummy", var="dummy")
selectDWD(res="dummy")
selectDWD(res="daily", per="r")
selectDWD(res="daily", var="kl")
selectDWD(id="01050", res=c("daily","monthly"), var="kl") # needs 'per'
selectDWD(id="00386", meta=TRUE)

selectDWD("Potsdam", res="daily", var="solar")
# should be an error:
berryFunctions::is.error( selectDWD(id="Potsdam", res="daily", var="solar"), TRUE)
berryFunctions::is.error( selectDWD(id="", current=TRUE) , tell=TRUE, force=TRUE)
```

Index

- *Topic **character**
 - findID, 7
 - rowDisplay, 16
- *Topic **chron**
 - readDWD, 15
- *Topic **datasets**
 - index, 8
 - metaInfo, 12
- *Topic **data**
 - dataDWD, 4
- *Topic **documentation**
 - rdwd, 14
- *Topic **file**
 - dataDWD, 4
 - dirDWD, 6
 - indexFTP, 9
 - readDWD, 15
 - selectDWD, 17
- *Topic **manip**
 - createIndex, 2
- *Topic **package**
 - rdwd, 14
- *Topic **spatial**
 - lldist, 11
- ==, 7, 18
- as.POSIXct, 16
- browseURL, 4
- climateGraph, 5
- createIndex, 2, 8, 9, 15, 19
- dataDWD, 2, 3, 4, 6, 15–17, 19
- dirDWD, 6
- download.file, 4, 5
- fileIndex, 2, 3, 10, 15, 18, 19
- fileIndex (index), 8
- findID, 7, 9, 15, 18
- fread, 16
- geoIndex, 2, 3, 17
- geoIndex (index), 8
- getURL, 9, 10
- getwd, 2, 4, 6, 10
- grepl, 7, 18
- index, 8
- indexFTP, 2, 3, 9, 9, 15, 19
- lldist, 11
- max, 11
- maxlldist (lldist), 11
- metaIndex, 2, 3, 7, 12, 13, 15, 18, 19
- metaIndex (index), 8
- metaInfo, 7, 9, 12
- monthAxis, 5
- nearbyStations, 12, 15
- newFilename, 5, 10
- popleaf, 16
- print, 12
- rdwd, 14
- rdwd-package (rdwd), 14
- read.fwf, 15
- readDWD, 4, 5, 15, 15
- rowDisplay, 16
- selectDWD, 3–5, 7–9, 13, 15, 16, 17
- setwd, 6
- strptime, 4, 16
- Sys.sleep, 4, 10
- unlist, 19