

Package ‘rdwd’

March 17, 2019

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

Version 1.0.0

Date 2019-03-17

Depends R(>= 2.10)

SystemRequirements Pandoc (>= 1.12.3), pandoc-citeproc

Imports berryFunctions (>= 1.17.0), pbapply

Suggests RCurl, leaflet, knitr, rmarkdown, testthat, data.table, OSMscale, R.rsp, raster, R.utils

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 6.1.1

VignetteBuilder R.rsp

BugReports <https://github.com/brry/rdwd/issues>

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-17 22:10:03 UTC

R topics documented:

createIndex	2
dataDWD	4

DEU	6
dirDWD	7
dwdbase	7
findID	8
index	9
indexFTP	10
lldist	11
metaInfo	12
nearbyStations	13
parameter_abbreviations	14
rdwd	15
readDWD	15
readMeta	17
readVars	18
rowDisplay	19
selectDWD	19

Index 23

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 = dwdbase, 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 <code>indexFTP</code> called with the same base value as this function
base	Main directory of DWD ftp server, defaulting to observed climatic records. DEFAULT: <code>dwdbase: ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate</code>
dir	Char: writeable directory name where to save the main output(s). Created if not existent. DEFAULT: "DWDdata" at current <code>getwd()</code>

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. Passed 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
```

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, base = dwdbase, dir = "DWDdata", force = FALSE,
  overwrite = FALSE, sleep = 0, quiet = FALSE, progbar = !quiet,
  browse = FALSE, read = TRUE, meta = grepl(".txt$", file),
  fread = FALSE, format = NA, ntrunc = 2, dfargs = NULL, ...)
```

Arguments

file	Char (vector): complete file URL(s) (including base and filename.zip) as returned by <code>selectDWD</code> . Can be a vector with several filenames.
base	Single char: base URL that will be removed from output file names. DEFAULT: <code>dwdbase: ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate</code>
dir	Char: Writeable directory name where to save the downloaded file. Created if not existent. DEFAULT: "DWDdata" at current <code>getwd()</code>
force	Logical (vector): always download, even if the file already exists in dir? Use NA to force re-downloading files older than 24 hours. If FALSE, it is still read (or name returned). DEFAULT: FALSE
overwrite	Logical (vector): if force=TRUE, overwrite the existing file rather than append "_1"/"_2" etc to the filename? DEFAULT: FALSE
sleep	Number. If not 0, a random number of seconds between 0 and sleep is passed to <code>Sys.sleep</code> 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 <code>length(file)==1</code> , progbar is internally set to FALSE. Only works if the R package pbapply is available. DEFAULT: TRUE (!quiet)
browse	Logical: open repository via <code>browseURL</code> 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 <code>readDWD</code> ? If FALSE, only download is performed and the filename(s) returned. DEFAULT: TRUE
meta	Logical (vector): is the file a meta file? Passed to <code>readDWD</code> . DEFAULT: TRUE for each file ending in ".txt"
fread	Fast reading? See <code>readDWD</code> . DEFAULT: FALSE

format	Char (vector): format used in <code>strptime</code> to convert date/time column, see <code>readDWD</code> . DEFAULT: NA
ntrunc	Single integer: number of filenames printed in messages before they get truncated with message "(and xx more)". DEFAULT: 2
dfargs	Named list of additional arguments passed to <code>download.file</code>
...	Further arguments passed to <code>readDWD</code> . Were passed to <code>download.file</code> prior to <code>rdwd 0.11.7 (2019-02-25)</code>

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 FTP files for a given station name and file path:
link <- selectDWD("Fuerstenzell", res="hourly", var="wind", per="recent")
# download file:
fname <- dataDWD(link, dir=tempdir(), read=FALSE) ; fname
# dir="DWDdata" is the default directory to store files
# unless force=TRUE, already obtained files will not be downloaded again

# read and plot file:
wind <- readDWD(fname) ; head(wind)
metafiles <- readMeta(fname) ; str(metafiles, max.level=1)
column_names <- readVars(fname) ; head(column_names)

plot(wind$MESS_DATUM, wind$F, main="DWD hourly wind Fuerstenzell", col="blue",
     xaxt="n", las=1, type="l", xlab="Date", ylab="Hourly Wind speed [m/s]")
berryFunctions::monthAxis(1, ym=T)

# current and historical files:
link <- selectDWD("Potsdam", res="daily", var="kl", per="hr"); link
potsdam <- dataDWD(link, dir=tempdir())
potsdam <- do.call(rbind, potsdam) # this will partly overlap in time
plot(TMK~MESS_DATUM, data=tail(potsdam,1500), type="l")
```

```
# The straight line marks the jump back in time
# Keep only historical data in the overlap time period:
potsdam <- potsdam[!duplicated(potsdam$MESS_DATUM),]

# With many files (>>50), use sleep to avoid getting kicked off the FTP server
#links <- selectDWD(res="daily", var="solar")
#sol <- dataDWD(links, sleep=20) # random waiting time after download (0 to 20 secs)

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

## End(Not run)
```

DEU	<i>Map of German states (Bundeslaender) from GADM through the raster package</i>
-----	--

Description

Map of German states (Bundeslaender) from GADM through the raster package

Format

Formal class 'SpatialPolygons' [package "sp"] with 4 slots

Details

Obtained with the code:

```
DEU1 <- raster::getData("GADM", country="DEU", level=1)
DEU <- rgeos::gSimplify(DEU1, tol=0.02, topologyPreserve=FALSE)
raster::plot(DEU1)
raster::plot(DEU)
save(DEU, file="data/DEU.rda")
tools::resaveRdaFiles("data/DEU.rda")
```

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, May 2018

dirDWD	<i>directory management for rdwd</i>
--------	--------------------------------------

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

See Also

[dataDWD](#)

Examples

```
# see source code of dataDWD and metaDWD
```

dwdbase	<i>DWD FTP Server base URL</i>
---------	--------------------------------

Description

base URL to DWD FTP Server / observed climatic records.
ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate

Usage

```
dwdbase
```

Format

An object of class character of length 1.

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: rdwd::metaIndex

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

```
# 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
```

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 243k rows x 8 columns:

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

metaIndex: data.frame with ca 82k 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 / Climate 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. Here's an example:

```
gridindex <- indexFTP("grids_germany", "ftp://ftp-cdc.dwd.de/pub/CDC")
gridindex <- indexFTP(gridindex, "ftp://ftp-cdc.dwd.de/pub/CDC", sleep=1)
```

Usage

```
indexFTP(folder = "currentfindex", base = dwdbase,
  is.file.if.has.dot = TRUE, 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. <code>"/hourly/wind/"</code> . Use <code>folder=""</code> to search at the location of base itself. If <code>codefolder</code> is <code>"currentfindex"</code> (the default) and base is the default, <code>codefolder</code> is changed to all folders in current <code>fileIndex</code> : <code>unique(dirname(fileIndex\$path))</code> . Leading slashes will be removed. DEFAULT: <code>"currentfindex"</code>
base	Main directory of FTP server, defaulting to DWD observed climatic records. Trailing slashes will be removed. DEFAULT: <code>dwdbase</code> : ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate
is.file.if.has.dot	Logical: if some of the input paths contain a dot, treat those as files, i.e. do not try to read those as if they were a folder. Only set this to FALSE if you know what you're doing. DEFAULT: TRUE
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: <code>"DWDdata"</code> at current <code>getwd()</code>

filename	Character: Part of output filename. "INDEX_of_DWD_" is prepended, "/" replaced with "_", ".txt" appended. DEFAULT: folder[1]
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 pbapply 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", dir=tempdir())
head(sol)

# mon <- indexFTP(folder="/monthly/kl", dir=tempdir(), verbose=TRUE)

## End(Not run)
```

lldist

distance between lat-long coordinates

Description

Great-circle distance between points at lat-long coordinates. Mostly a copy of `OSMscale::earthDist` Version 0.5.3 (2017-04-19). <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, <berry-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

id	Station ID (integer number or convertible to one)
hasfileonly	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	<i>Find DWD stations close to given coordinates</i>
----------------	---

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 YYYYMM-MDD, 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 location. 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

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

# see the use cases vignette for a continued example of this:
vignette("cases")
```

parameter_abbreviations

Parameter abbreviations for data on the DWD CDC FTP server

Description

Parameter abbreviations for data on the DWD CDC FTP server

Usage

```
parameter_abbreviations
```

Format

An object of class `data.frame` with 118 rows and 2 columns.

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Jun 2018

See Also

[readVars](#), [readDWD](#)

Examples

```
head(parameter_abbreviations)
```

rdwd

Download Climate Data from DWD (German Weather Service)

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](#). For an overview of available data, see `selectDWD`.

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](#), [noaa](#)
World data: [Global Surface Summary of the Day](#)
Durch data: <https://github.com/bvhest/KNMIr>
Canadian data: <https://cran.r-project.org/package=rclimateca>

readDWD

Process data from the DWD CDC FTP Server

Description

Read climate 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 = grepl(".txt$", file), binary = grepl(".tar.gz$",
  file), raster = grepl(".asc.gz$", file),
  multia = grepl("Standort.txt$", file), fread = FALSE, format = NA,
  tz = "GMT", progbar = TRUE, ...)
```

Arguments

file	Char (vector): name(s) of the file(s) downloaded with dataDWD , 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 (Beschreibung.txt)? DEFAULT: TRUE for each file ending in ".txt"
binary	Logical (vector): is the file a binary file, like for the radolan grid files? The argument selection (DEFAULT: NULL) can be used to read only a subset of the ~24*31=744 files (called as f[selection]). DEFAULT: TRUE for each file ending in ".tar.gz"
raster	Logical (vector): is the file a raster file, like for the seasonal grid files? The argument dividebyten (DEFAULT: TRUE) can be used to divide the values by ten. DEFAULT: TRUE for each file ending in ".asc.gz"
multia	Logical (vector): is the file a multi_annual file? Overrides meta, so set to FALSE manually if meta reading needs to be called on a file ending with "Standort". DEFAULT: TRUE for each file ending in "Standort.txt"
fread	Logical: read faster with data.table::fread? For 30 daily/kl/hist files, 7 instead of 10 seconds. NA can also be used, which means TRUE if data.table is available. DEFAULT: FALSE
format	Char (vector), only used if meta=FALSE: Format passed to as.POSIXct (see strptime) to convert the date/time column to POSIX time format. If NULL, no conversion is performed (date stays a factor). If NA, readDWD tries to find suitable format based on the number of characters. DEFAULT: NA
tz	Char (vector): time zone for as.POSIXct . "" 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 length(file)==1, progbar is internally set to FALSE. DEFAULT: TRUE
...	Further arguments passed to the underlying reading function, i.e. read.table , data.table::fread , read.fwf , readBin or raster::raster

Value

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

Author(s)

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

See Also

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

Examples

```
# see dataDWD
```

readMeta	<i>Process data from the DWD CDC FTP Server</i>
----------	---

Description

Read climate meta info textfiles in zip folders downloaded with [dataDWD](#).

Usage

```
readMeta(file, progbar = TRUE, ...)
```

Arguments

file	Char (vector): name(s) of the zip file(s) downloaded with dataDWD , e.g. "~/DWD-data/tageswerte_KL_02575_akt.zip"
progbar	Logical: present a progress bar with estimated remaining time? If missing and <code>length(file)==1</code> , progbar is internally set to FALSE. DEFAULT: TRUE
...	Further arguments passed to read.table

Value

Invisible named list of data.frames; or a list of lists, if `length(file)>1`.

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, 2016 + March 2019

See Also

[dataDWD](#), [readVars](#), [readDWD](#)

Examples

```
# see dataDWD
```

readVars	<i>Process data from the DWD CDC FTP Server</i>
----------	---

Description

Read climate variables (column meta data) from files downloaded with [dataDWD](#). The metadata file is read, processed and returned as a data.frame.
file can be a vector with several filenames.

Usage

```
readVars(file, progbar = TRUE)
```

Arguments

file	Char (vector): name(s) of the file(s) downloaded with dataDWD , e.g. "~/DWD-data/tageswerte_KL_02575_akt.zip"
progbar	Logical: present a progress bar with estimated remaining time? If missing and length(file)==1, progbar is internally set to FALSE. DEFAULT: TRUE

Value

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

Author(s)

Berry Boessenkool, <berry-b@gmx.de>, Jun 2018

See Also

[dataDWD](#), [readDWD](#), [parameter_abbreviations](#)

Examples

```
# see dataDWD
```

rowDisplay	<i>Create leaflet map popup from data.frame rows</i>
------------	--

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 character strings, one for each row in `x`.

Author(s)

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

See Also

[geoIndex](#)

selectDWD	<i>Select data from the DWD CDC FTP Server</i>
-----------	--

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 if `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 behavior in each case of availability:

case	<code>id</code>	<code>path</code>	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 2, you can explicitly set `res=""`, `var=""`, `per=""` to avoid the default interactive selection. 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/

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

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

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

Please note that both `"solar" (-/<<)` and `"sun" (<)` are available!

res=10_minutes	res=hourly	res=daily	res=monthly	res=annual
var=		kl <	kl <	kl <
		more_precip <	more_precip <	more_precip <
air_temperature <<	air_temperature <			
extreme_temperature <<				
extreme_wind <<				
	cloudiness <			
	cloud_type <			
precipitation <<	precipitation <			
	pressure <			
	soil_temperature <	soil_temperature <		
solar <<	solar -	solar -		
	sun <			
	visibility <			
		water_equiv <		
wind <<	wind <			

Please note that `1_minute/precipitation/historical` has subfolders for each year.

res=1_minute	res=multi_annual	res=subdaily
var=		
precipitation <<		
	mean_61-90 -	
	mean_71-00 -	
	mean_81-10 -	
		air_temperature <
		pressure <
		standard_format -

Usage

```
selectDWD(name = "", res = NA, var = NA, per = NA,
  exactmatch = TRUE, mindex = metaIndex, id = findID(name, exactmatch
```

```
= exactmatch, mindex = mindex), base = dwdbase, findex = fileIndex,
current = FALSE, meta = FALSE, meta_txt_only = TRUE,
outvec = any(per %in% c("rh", "hr")), ...)
```

Arguments

name	Char: station name(s) passed to <code>findID</code> , along with <code>exactmatch</code> and <code>mindex</code> . All 3 arguments are ignored if <code>id</code> is given. DEFAULT: ""
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: NA for interactive selection
var	Char: weather variable of interest, like e.g. "air_temperature", "cloudiness", "precipitation", See above and in <code>View(rdwd:::fileIndex)</code> . DEFAULT: NA for interactive selection
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: NA for interactive selection
exactmatch	Logical passed to <code>findID</code> : match name with <code>==</code> ? 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, exactmatch, mindex)</code>
base	Single char: main directory of DWD ftp server, defaulting to observed climatic records (<code>dwdbase</code>). Must be the same base used to create <code>findex</code> . DEFAULT: <code>ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate</code>
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: FALSE
meta	Logical: return metadata txt file name instead of climate data zip file? Relevant only in case 4 (path and id given) and case 3 for <code>res="multi_annual"</code> . See <code>metaIndex</code> for a compilation of all metaData files. DEFAULT: FALSE
meta_txt_only	Logical: if <code>meta</code> , only return .txt files, not the pdf and html files? DEFAULT: TRUE
outvec	Single logical: if path or ID length > 1, instead of a list, return a vector? (via <code>unlist</code>). DEFAULT: <code>per %in% c("rh", "hr")</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):
selectDWD("Potsdam", res="daily", var="kl", per="historical")

# all files for all stations matching "Koeln":
selectDWD("Koeln", res="", var="", per="", exactmatch=FALSE)
findID("Koeln", FALSE)

## Not run: # Excluded from CRAN checks to save time

# selectDWD("Potsdam") # interactive selection of res/var/per

# directly give station ID, can also be id="00386" :
selectDWD(id=386, res="daily", var="kl", per="historical")

# period can be abbreviated:
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="rh") # list if outvec=F
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), res="",var="",per="")
# all zip files in a given path (if ID is empty):
head( selectDWD(id="", res="daily", var="kl", per="recent") )

## End(Not run)
```

Index

- *Topic **character**
 - findID, 8
 - rowDisplay, 19
- *Topic **chron**
 - readDWD, 15
- *Topic **datasets**
 - DEU, 6
 - dwdbase, 7
 - index, 9
 - metaInfo, 12
 - parameter_abbreviations, 14
- *Topic **data**
 - dataDWD, 4
- *Topic **documentation**
 - rdwd, 15
- *Topic **file**
 - dataDWD, 4
 - dirDWD, 7
 - indexFTP, 10
 - readDWD, 15
 - readMeta, 17
 - readVars, 18
 - selectDWD, 19
- *Topic **manip**
 - createIndex, 2
- *Topic **package**
 - rdwd, 15
- *Topic **spatial**
 - lldist, 11
- ==, 8, 21
- as.POSIXct, 16

- browseURL, 4

- climateGraph, 5
- createIndex, 2, 9, 15, 21

- dataDWD, 3, 4, 7, 15–19, 22
- DEU, 6

- dirDWD, 7
- download.file, 5
- dwdbase, 2, 4, 7, 10, 21

- fileIndex, 2, 3, 10, 11, 15, 21
- fileIndex (index), 9
- findID, 8, 9, 15, 21
- fread, 16

- geoIndex, 2, 3, 19
- geoIndex (index), 9
- getURL, 10, 11
- getwd, 2, 4, 7, 10
- grepl, 8, 21

- index, 9
- indexFTP, 2, 3, 9, 10, 15, 21

- lldist, 11

- max, 12
- maxlldist (lldist), 11
- metaIndex, 2, 3, 8, 13–15, 21, 22
- metaIndex (index), 9
- metaInfo, 8, 9, 12
- monthAxis, 5

- nearbyStations, 13, 15
- newFilename, 5, 11

- parameter_abbreviations, 14, 18
- popleaf, 19
- print, 13

- raster, 16
- rdwd, 15
- rdwd-package (rdwd), 15
- read.fwf, 15, 16
- read.table, 16, 17
- readBin, 16
- readDWD, 4, 5, 14, 15, 15, 17, 18

readMeta, [16](#), [17](#)
readVars, [14](#), [16](#), [17](#), [18](#)
rowDisplay, [19](#)

selectDWD, [3–5](#), [8](#), [9](#), [13–16](#), [19](#)
setwd, [7](#)
strptime, [5](#), [16](#)
Sys.sleep, [4](#), [10](#)

unlist, [21](#)