Package 'MSCr'

February 19, 2018

Title Downloads and reads MSC data

Version 2.2.0

Date 2018-02-19

Author Kevin Shook, Centre for Hydrology, University of Saskatchewan

Maintainer Kevin Shook <kevin.shook@usask.ca>

Depends R (>= 3.3)

Imports CRHMr(>= 2.2.9), lubridate(>= 1.3), utils, stats, stringr

Description Functions to download and read in MSC data files.

License GPL-3

URL www.usask.ca/hydrology

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

MSCr-package   *Functions to read Meteorological Service of Canada (MSC) data*

Description

This package contains functions to do the following:

1. Read data from the MSC website and write CRHM files of hourly and daily observations.

2. Read data files of Adjusted and Homogenized Canadian Climate Data (AHCCD) monthly and daily data.

3. Read in file of MSC radar data and write a grid or xyz file

4. Read in large MSC data files containing daily or hourly values for many locations and write a CRHM obs file for each location.

5. Read in old format files of MSC or AES data for variables for a single site and write a CRHM obs file.

References

To cite **MSCr** in publications, use the command `citation('MSCr')` to get the current version of the citation.
The CRHM program is described in:
Pomeroy, John W, D M Gray, T Brown, N Hedstrom, W L Quinton, R J Granger, and S K Carey. 2007. "The Cold Regions Hydrological Model : A Platform for Basing Process Representation and Model Structure on Physical Evidence". Hydrological Processes 21 (19): 2650-2567.
The CRHM model may be downloaded from <http://www.usask.ca/hydrology/CRHM.php>.

---

bigMSCdailyToObs   *Reads large MSC files of daily values*

Description

Reads large MSC files holding daily values of several variables at several sites and exports an hourly CRHM obs data file for each site. The obs files are of the form '<sitenumber>_daily.obs''

Usage

```
bigMSCdailyToObs(infile, quiet = TRUE, logfile = "")
```

Arguments

`infile`	Required. Name of the file to be read.
`quiet`	Optional.Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave `quiet=TRUE` (i.e. the default). If you are working interactively, you will probably want to set `quiet=FALSE`.
`logfile`	Optional. Optional. Name of the file to be used for logging the action. Normally not used.

Value

If successful, returns TRUE. If unsuccessful, returns the value FALSE.

Author(s)

Kevin Shook

See Also

[bigMSChourlyToObs](#)

Examples

```
## Not run:
bigMSCdailyToObs('GRPextractor_PHW_Bad_Lake_dlyv2_21032015_155647.txt', quiet=FALSE)
## End(Not run)
```

---

bigMSChourlyToObs	*Reads large MSC files of hourly values*

Description

Reads large MSC files holding hourly values of several variables at several sites and exports an hourly CRHM obs data file for each site. The obs files are of the form '`<siten umber>_hourly.obs`''

Usage

```
bigMSChourlyToObs(infile, timezone = "", quiet = TRUE, logfile = "")
```

Arguments

`infile`	Required. Name of the file to be read.
`timezone`	Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under

Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time.

quiet   Optional. ptional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE.

logfile   Optional. Optional. Name of the file to be used for logging the action. Normally not used.

Value

If successful, returns TRUE. If unsuccessful, returns the value FALSE.

Author(s)

Kevin Shook #' @seealso bigMSCdailyToObs

Examples

```
## Not run:
bigMSChourlyToObs('GRPextractor_PHW_Bad_Lake_hly_21032015_152943.txt', timezone='CST', quiet=FALSE)
## End(Not run)
```

---

downloadMSCobs   *Downloads monthly and daily MSC data and creates CRHM .obs files*

Description

Downloads MSC hourly and daily data, one month at a time. The downloaded data are stored as .csv files, which are erased after the function terminates. Information about the .obs files is displayed on the screen and is also written to log files in the working directory. The obs files are: 1) hourly t, rh, and u 2) daily tmin, tmax, tmean and 3) daily precipitation. Because Environment Canada's web data is not consistent in any way, it is possble that using this function will generate warning messages.

Usage

```
downloadMSCobs(station.name = "", station.number = "", startyear = 1900,
  endyear = 2000, timezone = "", quiet = TRUE, logfile = "")
```

Arguments

station.name	Name of the station as a text string. This is used to create a directory that will hold the downloaded files and the created .obs files. It is also the basis for the names of the .obs files: '<station.name>Hourly.obs', '<station.name>DailyTemps.obs', and '<station.name>DailyPrecips.obs'.
station.number	Required,
startyear	Optional. First year for downloading. Default is 1900.
endyear	Optional. Last year for downloading. Default is 2000.
timezone	Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time.
quiet	Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE.
logfile	Optional. Optional. Name of the file to be used for logging the action. Normally not used.

Value

If successful, returns TRUE. If unsuccessful, returns the value FALSE.

Author(s)

Kevin Shook

References

The code for downloading MSC data is taken from http://www.fromthebottomoftheheap.net/2015/01/14/harvesting-canadian-climate-data. Some modifications were made to remove bad characters, and to download daily values.

Examples

```
## Not run:
downloadMSCobs('Vegreville', 1977, 1995, 1996, 'MST')
## End(Not run)
```

---

downloadMSCstations   *Reads current set of MSC stations from server*

Description

Reads current set of MSC stations from server

Usage

```
downloadMSCstations(stationURL = "", outfile = "", quiet = TRUE,
  logfile = "")
```

Arguments

stationURL	Optional. URL of file containing stations. If not specified then the file at "client_climate@ftp.tor.ec.gc.ca" will be used.
outfile	Optional. If specified, then the stations will be written to a comma delimited file.
quiet	Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE. Note that setting quiet=FALSE shows the downloading progress bar.
logfile	Optional. Name of the file to be used for logging the action. Normally not used.

Value

If unsuccessful, returns FALSE. If successful, returns a **CRHMr** data frame containing the station names and their metadata.

Note

The station data frame contains the variables: Name, Province, Climate ID, Station ID, WMO ID, TC ID, Latitude (Decimal Degrees), Longitude (Decimal Degrees), Latitude, Longitude, Elevation (m), First Year, Last Year, HLY First Year, HLY Last Year, DLY First Year, DLY Last Year, MLY First Year, MLY Last Year.

Author(s)

Kevin Shook

See Also

[downloadMSCobs](#)

Examples

```
## Not run:
MSCstations <- downloadMSCstations(quiet=FALSE)
## End(Not run)
```

---

oldMSCdailyPtoObs   *Creates an obs file from all MSC old-style precipitation files in a di-*
                                 rectory

Description

Reads all files of old-style MSC daily precipitation data from a specified directory and assembles
them. It is assumed is that only days with precipitaion were recorded. Writes a obs file called
'ppt.obs'.

Usage

```
oldMSCdailyPtoObs(directory = ".", timezone = "")
```

Arguments

directory   Optional. Directory containing all precipitation files. File names must begin
               with 'A00'. The default is the current directory.

timezone   Required. Required. The name of the timezone of the data as a character string.
               This should be the timezone of your data, but omitting daylight savings time.
               Note that the timezone code is specific to your OS. To avoid problems, you
               should use a timezone without daylight savings time. Under Linux, you can use
               'CST' and 'MST' for Central Standard or Mountain Standard time, respectively.
               Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Cen-
               tral Standard and Mountain Standard time. DO NOT use 'America/Regina' as
               the time zone, as it includes historical changes between standard and daylight
               savings time.

Value

If successful returns TRUE. If unsuccessful returns FALSE.

Examples

```
## Not run:
oldMSCdailyPtoObs(timezone='Etc/GMT+7')

## End(Not run)
```

---

oldMSCdailyTtoObs   *Creates an obs file from all MSC old-style daily temperature files in a*
                             directory

Description

Reads all files of old-style MSC daily soil, min and max air temp. data from a specified directory
and assembles them. The values are organized in columns, 1 year per file. Writes a obs file called
'xx_tminmax.obs', where 'xx' is the first 2 characters of 'filespec'.

Usage

```
oldMSCdailyTtoObs(directory = ".", filespec = "A1*", timezone = "")
```

Arguments

directory   Optional. Directory containing all temperature files. The default is the current
              directory.

filespec   Optional. File specification for all files. Default is 'A1*'.

timezone   Required. Required. The name of the timezone of the data as a character string.
              This should be the timezone of your data, but omitting daylight savings time.
              Note that the timezone code is specific to your OS. To avoid problems, you
              should use a timezone without daylight savings time. Under Linux, you can use
              'CST' and 'MST' for Central Standard or Mountain Standard time, respectively.
              Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Cen-
              tral Standard and Mountain Standard time. DO NOT use 'America/Regina' as
              the time zone, as it includes historical changes between standard and daylight
              savings time.

Value

If successful returns TRUE. If unsuccessful returns FALSE.

Examples

```
## Not run:
oldMSCHourlyUtoObs(timezone='Etc/GMT+7')

## End(Not run)
```

---

oldMSChourlyRHtoObs   *Creates an obs file from all MSC old-style RH files in a directory*

Description

Reads all files of old-style MSC hourly RH data from a specified directory and assembles them. The values are organized day x hour, 1 year per file. Writes a obs file called 'rh.obs'.

Usage

```
oldMSChourlyRHtoObs(directory = ".", timezone)
```

Arguments

directory   Optional. Directory containing all RH files. File names must begin with 'RH'. The default is the current directory.

timezone   Required. Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time.

Value

If successful returns TRUE. If unsuccessful returns FALSE.

Examples

```
## Not run:
oldMSChourlyRHtoObs(timezone='Etc/GMT+7')

## End(Not run)
```

---

oldMSChourlyTtoObs   *Creates an obs file from all MSC old-style temperature files in a directory*

Description

Reads all files of old-style MSC hourly temperature data from a specified directory and assembles them. The values are organized day x hour, 1 year per file. Writes an obs file called 't.obs'.

Usage

```
oldMSChourlyTtoObs(directory = ".", timezone)
```

Arguments

directory   Optional. Directory containing all temperature files. File names must begin with
                 `TE`. The default is the current directory.

timezone   Required. Required. The name of the timezone of the data as a character string.
                 This should be the timezone of your data, but omitting daylight savings time.
                 Note that the timezone code is specific to your OS. To avoid problems, you
                 should use a timezone without daylight savings time. Under Linux, you can use
                 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively.
                 Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Cen-
                 tral Standard and Mountain Standard time. DO NOT use 'America/Regina' as
                 the time zone, as it includes historical changes between standard and daylight
                 savings time.

Value

If successful returns `TRUE`. If unsuccessful returns `FALSE`.

Examples

```
## Not run:

oldMSChourlyTtoObs(timezone='Etc/GMT+7')

## End(Not run)
```

---

oldMSCHourlyUtoObs   *Creates an obs file from all MSC old-style wind files in a directory*

Description

Reads all files of old-style MSC hourly wind speed data from a specified directory and assembles
them. The values are organized day x hour, 1 year per file. Writes a obs file called 'u.obs'.

Usage

```
oldMSCHourlyUtoObs(directory = ".", timezone = "")
```

Arguments

directory   Optional. Directory containing all wind files. File names must begin with
                 'WIND'. The default is the current directory.

timezone Required. Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time.

Value

If successful returns TRUE. If unsuccessful returns FALSE.

Examples

```
## Not run:
oldMSCHourlyUtoObs(timezone='Etc/GMT+7')

## End(Not run)
```

| readAESdailyP | *Reads individual AES files of daily precipitation data in a directory and creates obs files.* |

Description

Reads very old Atmospheric Environment Service (AES) files. All of the files specified are assembled to create a CRHM observation file named 'ppt.obs' in the same directory as the data files. It is assumed that the files only contain days with recorded precipitation values, so days with zero values are inserted.

Usage

```
readAESdailyP(directory = ".", filespec = "A0O*", timezone = "")
```

Arguments

directory Optional. Directory containing files. If not specified, defaults to current directory. Note that this is an R path, which uses the '/' symbol on ALL operating systems.

filespec Optional. File specification (including wildcards) of the precipitation data. Default is ''A0O*''.

timezone Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST'

and 'MST' for Central Standard or Mountain Standard time, respectively. Under
Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Stan-
dard and Mountain Standard time. DO NOT use 'America/Regina' as the time
zone, as it includes historical changes between standard and daylight savings
time.

Value

If successful, returns TRUE. If unsuccessful, returns FALSE

Author(s)

Kevin Shook

See Also

readAEShourlyRH readAEShourlyWind readAEShourlyT codereadAESdailyTminTmax

Examples

```
## Not run:
readAESdailyP('./HistoricalData', timezone='etc/GMT+6')

## End(Not run)
```

---

readAESdailyTminTmax   *Reads individual AES files of daily tmin and tmax data in a directory*
                         and creates obs files.

Description

Reads very old Atmospheric Environment Service (AES) files of daily minimum and maximum air
temperatures. The air temperatures may be interpolated to hourly values by reading in the .obs file
using the functions readObsFile and tMinMaxToHourly in **CRHMr**.

Usage

```
readAESdailyTminTmax(directory, filespec = "A2*", timezone = "")
```

Arguments

directory   Optional. Directory containing AES data files. If not specified, defaults to cur-
               rent directory. Note that this is an R path, which uses the '/' symbol on ALL
               operating systems.

filespec   Optional. File specification (including wildcards) of the tmin data. Default is
               ''A2*''.

timezone   Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time.

Value

If successful, returns TRUE. If unsuccessful, returns FALSE

Author(s)

Kevin Shook

See Also

[readAEShourlyRH readAEShourlyWind readAEShourlyT readAESdailyP](#)

Examples

```
## Not run:
readAESdailyTminTmax('./HistoricalData', timezone='etc/GMT+6')

## End(Not run)
```

---

readAEShourlyRH   *Reads individual AES files of hourly RH data in a directory and creates obs files.*

Description

Reads very old Atmospheric Environment Service (AES) files. The files are named RHXX, where XX is the last 2 digits of the year. All of the files are assembled to create a CRHM observation file named 'rh.obs' in the same directory as the data files.

Usage

```
readAEShourlyRH(directory = ".", timezone = "")
```

Arguments

directory	Optional. Directory containing AES data files. If not specified, defaults to current directory. Note that this is an R path, which uses the `'/'` symbol on ALL operating systems.
timezone	Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time.

Value

If successful, returns TRUE. If unsuccessful, returns FALSE

Author(s)

Kevin Shook

See Also

[readAEShourlyWind](#) [readAEShourlyT](#) code[readAESdailyTminTmax](#) [readAESdailyP](#)

Examples

```
## Not run:
readAEShourlyRH('./HistoricalData', timezone='etc/GMT+6')

## End(Not run)
```

---

readAEShourlyT	*Reads individual AES files of hourly air temperature data in a directory and creates obs files*

Description

Reads very old Atmospheric Environment Service (AES) files. The files are named TEXXYY, where XX and YY are the last 2 digits of the beginning and end years in the dataset. All of the files are assembled to create a CRHM observation file named 't.obs' in the same directory as the data files.

Usage

```
readAEShourlyT(directory = ".", timezone = "")
```

Arguments

directory   Optional. Directory containing AES data files. If not specified, defaults to current directory. Note that this is an R path, which uses the `'/'` symbol on ALL operating systems.

timezone   Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time.

Value

If successful, returns TRUE. If unsuccessful, returns FALSE

Author(s)

Kevin Shook

See Also

[readAEShourlyRH readAEShourlyWind readAESdailyTminTmax readAESdailyP](#)

Examples

```
## Not run:
readAEShourlyT('./HistoricalData', timezone='etc/GMT+6')

## End(Not run)
```

---

readAEShourlyWind   *Reads individual AES files of hourly wind data in a directory and creates obs files.*

Description

Reads very old Atmospheric Environment Service (AES) files. The files are named WINDXX, where XX is the last 2 digits of the year. All of the files are assembled to create a CRHM observation file named 'u.obs' in the same directory as the data files. The wind speeds are converted from km/h to m/s.

Usage

```
readAEShourlyWind(directory = ".", timezone = "")
```

Arguments

directory   Optional. Directory containing AES data files. If not specified, defaults to cur-
                 rent directory. Note that this is an R path, which uses the `'/'` symbol on ALL
                 operating systems.

timezone   Required. The name of the timezone of the data as a character string. This
                 should be the timezone of your data, but omitting daylight savings time. Note
                 that the timezone code is specific to your OS. To avoid problems, you should
                 use a timezone without daylight savings time. Under Linux, you can use 'CST'
                 and 'MST' for Central Standard or Mountain Standard time, respectively. Under
                 Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Stan-
                 dard and Mountain Standard time. DO NOT use 'America/Regina' as the time
                 zone, as it includes historical changes between standard and daylight savings
                 time.

Value

If successful, returns TRUE. If unsuccessful, returns FALSE

Author(s)

Kevin Shook

See Also

[readAEShourlyRH](#)

Examples

```
## Not run:
readAEShourlyWind('./HistoricalData', timezone='etc/GMT+6')

## End(Not run)
```

---

readAHCCDdailyPrecips   *Reads AHCCD daily precipitation values.*

Description

This program reads Adjusted and Homogenized Canadian Climate Data (AHCCD) data in a month
x day data file of rainfall, snowfall or total precipitation.

Usage

```
readAHCCDdailyPrecips(infile)
```

Arguments

infile   Required. Name of the file to be read.

Value

If successful, returns the values in a dataframe, consisting of the date, the value and the data code. If unsuccessful, returns the value FALSE.

Note

The AHCCD are often used for statistical analysis, so this function returns a data quality code, and does not create a CRHM obs file.

Author(s)

Kevin Shook

References

Monthly AHCCD data are available from <http://www.ec.gc.ca/dccha-ahccd>. Daily values must be requested. Any use of the precipitation data must cite *Mekis, E and L.A. Vincent, 2011: An overview of the second generation adjusted daily precipitation dataset for trend analysis in Canada. Atmosphere-Ocean, 49 (2), 163-177.*

See Also

readAHCCDdailyTemps readAHCCDmonthlyPrecips

Examples

```
## Not run:
stoon.snowfall <- readAHCCDdailyPrecips('3031093_S.txt')
## End(Not run)
```

---

readAHCCDdailyTemps   *Reads AHCCD daily temperature values*

Description

This program reads Adjusted and Homogenized Canadian Climate Data (AHCCD) data in a month x day data file of min max or mean air temperatures.

Usage

```
readAHCCDdailyTemps(infile)
```

Arguments

infile   Required. Name of the file to be read.

Value

If successful, returns the values in a dataframe, consisting of the date, the value and the data code. If unsuccessful, returns the value FALSE.

Note

The AHCCD are often used for statistical analysis, so this function returns a data quality code, and does not create a CRHM obs files.

Author(s)

Kevin Shook

References

Monthly AHCCD data are available from <http://www.ec.gc.ca/dccha-ahccd>. Daily values must be requested. Any use of the temperature data must cite *Vincent, L. A., X. L. Wang, E. J. Milewska, H. Wan, F. Yang, and V. Swail , 2012. A second generation of homogenized Canadian monthly surface air temperature for climate trend analysis, J. Geophys. Res., 117, D18110, doi:10.1029/2012JD017859.*

See Also

readAHCCDdailyTemps readAHCCDmonthlyPrecips

Examples

```
## Not run:
stoon.tmax <- readAHCCDdailyTemps('dx40657120.txt')
## End(Not run)
```

---

readAHCCDmonthlyPrecips

                    Reads AHCCD monthly precipitation values.

Description

This program reads Adjusted and Homogenized Canadian Climate Data (AHCCD) data in a month x day data file of rainfall, snowfall or total precipitation.

Usage

```
readAHCCDmonthlyPrecips(infile)
```

Arguments

infile   Required. Name of the file to be read.

Value

If successful, returns the values in a dataframe, consisting of the date, the value and the data code. If unsuccessful, returns the value FALSE.

Note

The AHCCD are often used for statistical analysis, so this function returns a data quality code, and does not create a CRHM obs file.

Author(s)

Kevin Shook

References

Monthly AHCCD data are available from `http://www.ec.gc.ca/dccha-ahccd`. Any use of the precipitation data must cite *Mekis, E and L.A. Vincent, 2011: An overview of the second generation adjusted daily precipitation dataset for trend analysis in Canada. Atmosphere-Ocean, 49 (2), 163-177.*

See Also

`readAHCCDdailyTemps readAHCCDdailyPrecips`

Examples

```
## Not run:
stoon.monthly.total <- readAHCCDmonthlyPrecips('mt4057120.txt')
## End(Not run)
```

---

readDatamartHourlyPrecip

Reads hourly Datamart-format precipitation data

Description

Reads hourly precipitation values from a file in the MSC Datamart format (day x hour).

Usage

```
readDatamartHourlyPrecip(infile, timezone = "")
```

Arguments

infile   Required. File to read precipitation from.

timezone   Optional. Timezone of the orginal data. If not specified, your timezone will be used.

Value

Returns a data frame consisting of the variables 'datetime', 'precip' and 'code'. The precipitation is in mm. The code is the data quality code from the original dataset. Missing values coded as -9999 in the original file are set to `NA_real_`.

Examples

```
## Not run:
precip <- readDatamartHourlyPrecip('L1012710.262')
## End(Not run)
```

---

readNOAAprecip   *Read NOAA file of precipitation data*

Description

Read NOAA file of precipitation data

Usage

```
readNOAAprecip(NOAAfile = "", outLoc = "./", timezone = "",
  quiet = TRUE, logfile = "")
```

Arguments

NOAAfile   Required. File of NOAA precipitation data. Contains PRECIP and/or SNOW-
                FALL variables. May contain data for any number of locations.

outLoc   Optional. Location for output files. Each variable for each location will be
                written to a separate **CRHMr** obs file.

timezone   Required. The name of the timezone of the data as a character string. This
                should be the timezone of your data, but omitting daylight savings time. Note
                that the timezone code is specific to your OS. To avoid problems, you should
                use a timezone without daylight savings time. Under Linux, you can use 'CST'
                and 'MST' for Central Standard or Mountain Standard time, respectively. Under
                Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Stan-
                dard and Mountain Standard time. DO NOT use 'America/Regina' as the time
                zone, as it includes historical changes between standard and daylight savings
                time.

quiet   Optional. Suppresses display of messages, except for errors. If you are calling
                this function in an R script, you will usually leave quiet=TRUE (i.e. the default).
                If you are working interactively, you will probably want to set quiet=FALSE.

logfile   Optional. Name of the file to be used for logging the action. Normally not used.

Value

If successful, returns `TRUE`. If unsuccessful, returns `FALSE`.

Examples

```
## Not run: result <- readNOAAprecip(NOAAfile = "4553527558696dat.txt", timezone = "MST")
```

---

readRadarPrecip   *Reads MSC radar precipitation*

Description

Reads in file containing MSC radar precipitation data and adds the missing (zero) values. The data consist of a header, followed by a single line containing the word 'DATA' and all of the values, which are comma-delimited.

Usage

```
readRadarPrecip(radarFile = "", outputFormat = "grid")
```

Arguments

radarFile   Required. File containing precipitation data

outputFormat   Optional. Either 'grid' (the default) or 'XYZ'.

Value

If successful, returns radar data in specified format, which consists of a list containing the header information and either grided (matrix) or XYZ (dataframe) values. The precipitation vaues are in mm/hr. If unsuccessful, returns FALSE.

Author(s)

Kevin Shook

Examples

```
## Not run:
a <- readRadarPrecip('junk.num')
## End(Not run)
```

---

singleACIStoObs   *Converts single ACIS file to obs files*

Description

Reads in a single file of met data downloaded from the Alberta Climate Information Service (ACIS), <https://agriculture.alberta.ca/acis/>. Each station's values are written to a separate `.obs` file, named after the station. The comments for all of the variables are written to a separate `.csv` file for each station.

Usage

```
singleACIStoObs(ACISfile = "", outDir = "", timezone = "etc/GMT+7",
  quiet = TRUE)
```

Arguments

ACISfile	Required. The file containing the ACIS data. Note that the file name should be the default produced by the ACID download system, and daily file names must contain the word 'Daily'.
outDir	Optional. The location for the `.obs` files. If not specified, then the directory containing the ACIS file will be used.
timezone	Optional. The name of the timezone of the data as a character string. This should be the timezone of the data, i.e. Mountain Standard Time. Note that the timezone code is specific to your OS. Under Linux, you can use 'MST' for Mountain Standard time. Under Windows or OSX, you can use 'etc/GMT+7', which is the default.
quiet	Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE.

Value

If successful, returns TRUE. If unsuccessful, returns FALSE.

Author(s)

Kevin Shook

Examples

```
## Not run:
result <- singleACIStoObs('ACISHourlyData-20170209-20170223-PID122537324.csv')
## End(Not run)
```

Index