Package 'geotopbricks'

July 2, 2014

Maintainer Emanuele Cordano <emanuele.cordano@gmail.com>

License GPL (>= 2)

Title geotopbricks

Type Package

Author Emanuele Cordano, Daniele Andreis, Fabio Zottele

Description geotopbricks: Analyzes raster maps and other information as input/output files from the Hydrological Distributed Model GEOtop. It contains functions and methods to import maps and other keywords from geotop.inpts file. Any information about the GEOtop Distributed Hydrological Model is available on www.geotop.org. The examples are tested on two simulation cases run with GEOtop built 1.225-9 mostly developed by Stefano Endrizzi. Bugs/comments/questions/collaboration of any kind are warmly welcomed.

Version 1.3.5.4

Date 2014-05-20

Depends R (>= 2.10),methods,raster,stringr,zoo

Suggests rgdal,soilwater

URL www.geotop.org,http://cri.fmach.eu/Research/Sustainable-Agro-Ecosystems-and-Bioresources/Dynamics-in-the-agro-ecosystems/people/Emanuele-Cordano

NeedsCompilation no

Repository CRAN

Date/Publication 2014-05-22 19:53:54

R topics documented:

`geotopbricks-package`

geotopbricks: Analyzes raster maps as input/output files from the Hydrological Distributed Model GEOtop

---

Description

This packages uses R raster utilities to read and analize outputs of the Distributed Hydrological Model GEOtop www.geotop.org. It contains functions and methods to import maps and other keywords from geotop.inpts file. Any information about the GEOtop Distributed Hydrological Model is available on www.geotop.org. Two examples are shown: http://meteogis.fmach.it/idroclima/panola13_run2xC_test3/ and http://meteogis.fmach.it/idroclima/ton-toss/. These examples are tested on two simulation cases run with GEOtop built 1.225-9 mostly developed by Stefano Endrizzi (http://www.geo.uzh.ch/en/units/physical-geography-3g/about-us/staff/stefano-endrizzi). Bugs/comments/questions/collaboration of any kind are warmly welcomed.

Details

Package:	geotopbricks
Type:	Package
Version:	1.3.5
Date:	2013-08-25
License:	GPL (>= 2)
LazyLoad:	yes
Depends:	zoo,rgdal,methods,stringr,raster,soilwater

Note

geotobricks is an on-going project. All criticism, comments and suggestions are well welcomed.

geotobricks is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

geotopbricks is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

Author(s)

Emanuele Cordano <emanuele.cordano@gmail.com>, Daniele Andreis, Fabio Zottele.

References

wwww.geotop.org

bondone *Bondene Dataset*

Description

It contains hourly meteorological data observed at MeteoTrentino T0327 station located at Monte Bondone-Viotte (Trentino, Easter Alps, Italy) from August 2004 to December 2012.\

The zoo object 'meteo' contains:

Iprec Hourly Precipitation Depth expressed in millimeters

AirT Air Temperature expressed in Celsius Degree

RH Relative Humidity in PerCent

WinDir Wind Direction expressed in Degrees North Clockwise

WinSp Wind Direction expressed in meters per second

Swglob Short-Wave Radiation expressed in Watts per square meters

The corresponding time axis vector for each observation can be printed by typing index(meteo).

Usage

```
data(bondone)
```

Format

Data frame , 'zoo' object

Details

This data set stores all meteorological information useful for a GEOtop www.geotop.org simulation. The user can easily use the package with his/her own data after replacing the values of such variables.

Source

Original data are provided by Provincia Autonoma di Trento (http://www.meteotrentino.it/)).

This dataset is intended for research purposes only, being distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY.

brick *brick*

Description

Added implemetation for 'brick' S4 method

brick method for GeotopRasterBrick

Usage

```
## S4 method for signature 'zoo'
brick(x, layer = 1, timerange = NULL, time = NULL,
  rows = 1:nrow(x), crs = NULL, use.read.raster.from.url = TRUE)

## S4 method for signature 'GeotopRasterBrick'
brick(x)
```

Arguments

x	a 'zoo' object returned by function pointer.to.maps.xyz.time or pointer.to.maps.xy.ti or a GeotopRasterBrick-class object
layer	layer at which raster maps are imported. If is NULL, maps ara no-zlayer distributed and zoo must be returend by pointer.to.maps.xy.time
timerange	two-elememts vector containing the time range at which geotop maps are imported
time	vector of time instants at which geotop maps are imported
rows	rows of zoo correspondig to the geotop maps that are imported. By default all rows of zoo are considered. It is calculated by time or timerange if they are not set as NULL.
crs	coordinate system see RasterBrick-class
use.read.raster.from.url	logical value. Default is TRUE. If TRUE the RasterLayer are read with read.raster.from.url, istead of raster (otherwise). It is recomended in case the files whose paths are contained in x are remote and are 'http' addresses. In this cases the stand-alone method raster(x) does not always work and use.read.raster.from.url is necessary.

Value

a RasterBrick-class containing the geopop maps indicated by x, which is already in a GeotopRasterBrick-class object or a 'zoo' object returned by function pointer.to.maps.xyz.time or pointer.to.maps.xy.time.

See Also

getvalues.brick.at.depth,vertical.aggregate.brick.within.depth

Examples

```
# TON TOSS
# See the examples in the functions listed in the 'SeeAlso' section
```

`brick.decimal.formatter` *Imports a brick of raster ascii maps into a 'brick' object*

Description

Imports a brick of raster ascii maps into a 'brick' object

Usage

```
brick.decimal.formatter(file = NULL, file_prefix, formatter = "%04d",
  file_extension = ".asc", nlayers = 10, use.read.raster.from.url = FALSE,
  crs = NULL, start.from.zero = FALSE)
```

Arguments

`file`	fileneme of the 'brick' files containing the decimal formatter. It is `NULL` by default, otherwise it replaces `file_suffix`, `formatter` and `file_extension`.
`file_prefix`	character string suffix name of the 'brick' files.
`formatter`	string value. Default is `"%04d"` .
`file_extension`	strinf value. Default is `".asc"`
`nlayers`	number of layers
`use.read.raster.from.url`	logical value. Default is `FALSE`. (this is recommended in this function). If `TRUE` the RasterLayer are read with `read.raster.from.url`, istead of `raster` (otherwise). It is recomended in case the files whose paths are contained in `x` are remote and are 'http' addresses. In this cases the stand-alone method `raster(x)` does not always work and `use.read.raster.from.url` is necessary.
`start.from.zero`	logical value. Default is `FALSE`. If `TRUE` the formatter starts from `0000`, otherwise it starts from `0001`.
`crs`	coordinate system see `RasterBrick-class`,`brick`, Default is `NULL`.

Value

the output is returned as a `RasterBrick-class` object

Examples

```
library(geotopbricks)
library(raster)
file <- system.file("doc/examples/snowthickness",package="geotopbricks")
file <- paste(file,"SnowThickness0000L%04d.asc",sep="/")
# nlayers=15
nlayers <- 6 ## Only 6 layers are read to minimize the elapsed time of the example!!
b <- brick.decimal.formatter(file=file,nlayers=nlayers)
nlayers(b)
names(b)
```

---

brickFromOutputSoil3DTensor

Extracts a brick or a raster layer from a output 3D Tensor or 2D map respectively

---

Description

Extracts a brick or a raster layer from a output 3D Tensor or 2D map respectively

Usage

```
brickFromOutputSoil3DTensor(x, when, layers = "SoilLayerThicknesses",
  one.layer = FALSE, suffix = "L%04dN%04d.asc", wpath = NULL,
  tz = "A", start_date_key = "InitDateDDMMYYYYhhmm",
  end_date_key = "EndDateDDMMYYYYhhmm", timestep = "OutputSoilMaps",
  use.read.raster.from.url = FALSE, crs = NULL, projfile = "geotop.proj",
  start.from.zero = FALSE, secondary.suffix = NULL, ...)

rasterFromOutput2DMap(x, when, ...)
```

Arguments

x	string. GEOtop keyword reletated to the 3D or 2D variable to be imported in R.
when	POSIXct-class for date and time on which the variable x is requested.
layers	number of soil layer or geotop keyword for soil leyer (e.g. SoilLayerThicknesses or SoilFile). Default is SoilLayerThicknesses.
timestep	time step expressed in seconds every which the raster file has been created. It can be a string corresponding to the geotop keyword in the inpts file. Default value is "OutputSoilMaps".
suffix	charcher string containing the decimal formatter used by GEOtop in the output file names. Default is "L not to modify the value of this argument and use the default value.
wpath,tz,use.read.raster.from.url	see get.geotop.inpts.keyword.value

projfile — name of the *.proj file containing CRS information. See get.geotop.inpts.keyword.value. Default is "geotop.proj". If is NULL or NA or this file does not exist, it is not searched and read.. In case use.read.raster.from.url is TRUE and no NULL or NA values are assinged, the *.proj file is searched.

crs,start.from.zero — see brick.decimal.formatter. If crs is not NULL (Default) , projfile is ignored.

one.layer — logical value. If TRUE a RasterLayer-class object is imported, otherwise a RasterBrick-classobject is returened. Default for brickFromOutputSoil3DTensor is FALSE

start_date_key,end_date_key — initial and final detes and times of the GEOtop simulation or alternatively the respective keywords of *.inpts file (Default)

secondary.suffix — String secondary suffix which can be added at the end of the Map file name (optional). Default is NULL and no secondary suffix is added.

... — additional arguments for get.geotop.inpts.keyword.value or brickFromOutputSoil3D

Details

These functions brickFromOutputSoil3DTensor and rasterFromOutput2DMap return 3D or 2D Raster-class objects respectively. rasterFromOutput2DMap is a wrapper function of brickFromOutputSoil3DTensor with the option one.layer==TRUE. The functionswork with the following output keywords:

"SoilTempTensorFile",

"SoilAveragedTempTensorFile",

"SoilLiqContentTensorFile",

"SoilAveragedLiqContentTensorFile",

"SoilIceContentTensorFile",

"SoilAveragedIceContentTensorFile",

"SoilLiqWaterPressTensorFile",

"SoilTotWaterPressTensorFile" for brickFromOutputSoil3DTensor;

"FirstSoilLayerTempMapFile",

"FirstSoilLayerAveragedTempMapFile",

"FirstSoilLayerLiqContentMapFile",

"FirstSoilLayerIceContentMapFile",

"LandSurfaceWaterDepthMapFile",

"ChannelSurfaceWaterDepthMapFile",

"NetRadiationMapFile",

"InLongwaveRadiationMapFile",

"NetLongwaveRadiationMapFile",

"NetShortwaveRadiationMapFile",

"InShortwaveRadiationMapFile",
"DirectInShortwaveRadiationMapFile",
"ShadowFractionTimeMapFile",
"SurfaceHeatFluxMapFile",
"SurfaceSensibleHeatFluxMapFile",
"SurfaceLatentHeatFluxMapFile",
"SurfaceTempMapFile",
"PrecipitationMapFile",
"CanopyInterceptedWaterMapFile",
"SnowDepthMapFile",
"GlacierDepthMapFile",
"SnowMeltedMapFile",
"SnowSublMapFile",
"GlacierMeltedMapFile",
"GlacierSublimatedMapFile",
"AirTempMapFile",
"WindSpeedMapFile",
"WindDirMapFile",
"RelHumMapFile",
"SWEMapFile",
"GlacierWaterEqMapFile"
"SnowDurationMapFile",
"ThawedSoilDepthMapFile",
"ThawedSoilDepthFromAboveMapFile",
"WaterTableDepthMapFile",
"WaterTableDepthFromAboveMapFile",
"NetPrecipitationMapFile",
"EvapotranspirationFromSoilMapFile" for rasterFromOutput2DMap.

Author(s)

Emanuele Cordano

See Also

get.geotop.inpts.keyword.value,brick.decimal.formatter

Examples

```
library(geotopbricks)
# The data containing in the link are only for educational use
wpath <- "http://www.boussinesq.org/geotopbricks/simulations/idroclim_test1"
x <- "SoilLiqContentTensorFile"
tz <-  "Etc/GMT+1"
when <- as.POSIXct("2002-03-22",tz=tz)

# Not Run because it elapses too long time!!!
# Please Uncomment the following lines to run by yourself!!!
# b <- brickFromOutputSoil3DTensor(x,when=when,wpath=wpath,tz=tz,use.read.raster.from.url=TR

# a 2D map:
x_e <- "SnowDepthMapFile"
# Not Run: uncomment the following line
# m <- rasterFromOutput2DMap(x_e,when=when,wpath=wpath,timestep="OutputSnowMaps",
#                            tz=tz,use.read.raster.from.url=TRUE)
## NOTE: set use.read.raster.from.url=FALSE (default)
# if the "wpath" directorty is in the local file system.
# Not Run: uncomment the following line
# plot(m)
```

`color.bar` *Graphic Representation of a Color bar, function written by John Colby*

Description

Graphic Representation of a Color bar, function written by John Colby

Usage

```
color.bar(lut, min, max = -min, nticks = 11, ticks = seq(min, max, len =
  nticks), title = "", width = 1.75, height = 5, ncolmax = 100,
  digits = 4, pdf = NULL)
```

Arguments

`lut`	see reference `http://stackoverflow.com/questions/9314658/colorbar-from-cust`
`min`	see reference `http://stackoverflow.com/questions/9314658/colorbar-from-cust`
`max`	see reference `http://stackoverflow.com/questions/9314658/colorbar-from-cust`
`nticks`	see reference `http://stackoverflow.com/questions/9314658/colorbar-from-cust`
`ticks`	see reference `http://stackoverflow.com/questions/9314658/colorbar-from-cust`
`title`	see reference `http://stackoverflow.com/questions/9314658/colorbar-from-cust`
`width,height`	width and height of the device
`digits`	specified number of significant digits
`pdf`	character value for pdf output file. Default is `NULL` and no pdf file is created.
`ncolmax`	maximum number of colors. Default is 100.

Note

This function is taken from http://stackoverflow.com/questions/9314658/colorbar-from-custom-c
Plese visit the URL for major details and give your feedback if possible.

Author(s)

John Colby http://stackoverflow.com/users/412342/john-colby

References

http://stackoverflow.com/questions/9314658/colorbar-from-custom-colorramppalette

Examples

```
color.bar(colorRampPalette(c("light green", "yellow", "orange", "red"))(100), -1)
```

---

color.bar.raster *Graphic Representation of a Color legend of a Raster or Geotopbrick-Raster object as a Color bar, inspired by the function written by John Colby*

Description

Graphic Representation of a Color legend of a Raster or GeotopbrickRaster object as a Color bar, inspired by the function written by John Colby

Usage

```
color.bar.raster(x, col, ...)
```

Arguments

x	a Rster or GeotopRasterBrick object
col	the color palette used
...	arguments to be passed to color.bar

See Also

color.bar

`create.geotop.inpts.keyword` *Creates an 'geotop.inpts' files the keyword and their values of a date.frame like the one returned by* `declared.geotop.inpts.keywords`

Description

Creates an 'geotop.inpts' files the keyword and their values of a date.frame like the one returned by `declared.geotop.inpts.keywords`

Usage

```
create.geotop.inpts.keyword(df, file = "geotop.inpts.copy", wpath = NULL,
  comment.lines = "default", header = "default", ...)
```

Arguments

`df`	data frame returend by `declared.geotop.inpts.keywords`
`file`	connetion or file name where to write 'df'
`wpath`	complere path to `file` (optional). Default is `NULL`.
`comment.lines`	string or vector of strings to add as comments for each keyword. If it is `NULL` the comment lines are omitted.
`header`	string or vector of strings to add as a header. If it is `NULL` the header is omitted.
`...`	further arguments for `writeLines`

Details

In case `comment.lines` and `header` are set equal to `"default"`, they are suitably modified within the function code. See the example output.

See Also

`writeLines`,`declared.geotop.inpts.keywords`

Examples

```
library(geotopbricks)

#Simulation working path
wpath <- 'http://www.boussinesq.org/geotopbricks/simulations/panola13_run2xC_test3'
df <- declared.geotop.inpts.keywords(wpath=wpath)
create.geotop.inpts.keyword(df=df)
```

`create.geotop.meteo.files` *Creates geotop meteo files from (a list of) 'zoo' objects*

Description

Creates geotop meteo files from (a list of) 'zoo' objects

Usage

```
create.geotop.meteo.files(x, format = "%d/%m/%Y %H:%M",
  file_prefix = "meteo", file_extension = ".txt", formatter = "%04d",
  na = "-9999", col.names = TRUE, row.names = FALSE,
  date_field = "Date", sep = ",", level = NULL, quote = FALSE, ...)
```

Arguments

`x`	'zoo' object or a list of 'zoo' object representing the meteorological station
`format`	string format representing the date, see `as.POSIXlt`. Default is `"%d/%m/%Y %H:%M"` (which is the same format used in `geotop.inpts` keyword `InitDateDDMMYYYYhhmm`)
`file_prefix`	string containing file prefix (full path). It correspos to the value of in `geotop.inpts` keyword `MeteoFile`)
`file_extension`	string containing the extensions of final files. Default is `c(".txt")`
`formatter`	string value. It is the decimal formatter contained in the file name and used in case the tabular data are referred at several points. Default is `"%04d"` . See `sprintf`.
`na`	NA value indicator. Default is `"-9999"`. See `write.table`.
`row.names`	logical parameter. Default is `FALSE`. See `write.table`.
`col.names`	logical parameter. Default is `TRUE`. See `write.table`.
`date_field`	string value. Default is "Date", otherwise defined by the value of `HeaderDateDDMMYYYYhhmmMeteo` geotop keyword.
`sep`	string value. Default is `","`. See `write.table`.
`quote`	logical parameter. Default is `TRUE`. See `write.table`.
`level`	integer argument. See `get.geotop.inpts.keyword.value` for major details. Default is `NULL` and is ignored.
`...`	further argurments for `write.table`

See Also

`write.table`,`get.geotop.inpts.keyword.value`

Examples

```
library(geotopbricks)
data(bondone)
## Not Run - Uncomment te following line to run the example
## create.geotop.meteo.files(x=meteo)
```

`declared.geotop.inpts.keywords`
Collects all keywords contained in the 'getop.inpts' configuration files and their values in a data frame object.

Description

Collects all keywords contained in the 'getop.inpts' configuration files and their values in a data frame object.

Usage

```
declared.geotop.inpts.keywords(wpath, inpts.file = "geotop.inpts",
  comment = "!", exceptions = "Date", warn = FALSE, ...)
```

Arguments

`wpath`	working directory containing GEOtop files
`inpts.file`	name of the GEOtop configuration file. Default is `"geotop.inpts"`
`comment`	comment indicator charcater. Default is `"!"`
`exceptions`	string vector. If keywords contain an element of this vector, the blank spaces in Value `"  "` will not be removed.
`warn`	logical argument of `readLines`. Default is `FALSE`.
`...`	further arguments of `readLines`

Value

a data frame with two columns: `Keyword` and `Value`

See Also

`get.geotop.inpts.keyword.value`

geotopbrick *geotopbrick*

Description

geotopbrick

geotopbrick method bla bla bla

Usage

```
geotopbrick(x = NULL, ...)

## Default S3 method:
geotopbrick(x, ...)

## S3 method for class 'zoo'
geotopbrick(x, layer = NULL, time = NULL, crs = NULL,
  timerange = NULL, ...)

## S3 method for class 'RasterLayer'
geotopbrick(x, layer = NULL, time = NULL,
  ascpath = zoo(NULL), ...)

## S3 method for class 'RasterBrick'
geotopbrick(x, layer = NULL, time = NULL,
  ascpath = zoo(NULL), ...)

## S3 method for class 'GeotopRasterBrick'
geotopbrick(x, layer = NULL, time = NULL,
  crs = NULL, timerange = NULL, ascpath = NULL, ...)
```

Arguments

x	a 'zoo' object returned by function pointer.to.maps.xyz.time or pointer.to.maps.xy.ti or a GeotopRasterBrick-class object
layer	layer at which raster maps are imported. If is NULL, maps ara no-zlayer distributed and zoo must be returend by pointer.to.maps.xy.time
time	vector of time instants at which geotop maps are imported
crs	coordinate system see RasterBrick-class
timerange	two-elememts vector containing the time range at which geotop maps are imported
ascpath	NULL object or a "zoo" S3 object containing the names of ascii maps provided by GEOtop
...	further arguments.

Value

a GeotopRasterBrick-class

GeotopRasterBrick-class
GeotopRasterBrick-class

Description

A GeotopRasterBrick: an object to manage raster maps provied by GEOtop!!

Details

ascpath: A "zoo" S3 object containing the names of ascii maps provided by GEOtop

index: A "POSIXt" S3 object containing time or dates on which raster layers of brick are referred

layer: character. Name of the vertical layer at which raster map are referred

brick: A "RasterBrick-class" S4 object containing the Raster-Layer maps imported from GEOtop output files

#'

Note

A GeotopRasterBrick object can be created by new("GeotopRasterBrick", ...)

Author(s)

Emanuele Cordano

See Also

Raster-class

Examples

```
showClass("GeotopRasterBrick")
```

get.geotop.inpts.keyword.value

Returns the values of a keyword of "geotop.inpts" file or data frame with the suitable format

Description

Returns the values of a keyword of "geotop.inpts" file or data frame with the suitable format

Usage

```
get.geotop.inpts.keyword.value(keyword, inpts.frame = NULL,
  vector_sep = NULL, numeric = FALSE, format = "%d/%m/%Y %H:%M",
  date = FALSE, tz = "Etc/GMT+1", raster = FALSE,
  file_extension = ".asc", add_wpath = FALSE, wpath = NULL,
  use.read.raster.from.url = TRUE, data.frame = FALSE,
  formatter = "%04d", level = 1, date_field = "Date", isNA = -9999,
  matlab.syntax = TRUE, projfile = "geotop.proj", start_date = NULL,
  end_date = NULL, ContinuousRecovery = 0,
  ContinuousRecoveryFormatter = "_crec%04d", ...)
```

Arguments

keyword	keyword name
inpts.frame	data frame returned by declared.geotop.inpts.keywords or NULL. Default is NULL.
vector_sep	character value for the separator chacter if Keyword Value must be returned as a vector, otherwise it is NULL. Default is NULL, but if numeric or date are FALSE, vector_sep is set "," by default.
numeric	logical value. If TRUE the Value has numeric type, otherwise it is a string or string vector. Default is FALSE.
date	logical value. If TRUE the Value is retured as POSIXlt date, otherwise it is a string or string vector. Default is FALSE.
format	string format representing the date, see as.POSIXlt, used if date is TRUE. Default is "%d/%m/%Y %H:%M" (which is the format used in geotop.inpts keyword InitDateDDMMYYYYhhmm)
tz	format string representing the time zone, see as.POSIXlt, used if date is TRUE. Default is "Etc/GMT+1" (until the previous version it was "A") which meens UTC +1.
raster	logical value. Default is FALSE. If TRUE function returns direclty the raster map as Raster-class object built with raster method.
file_extension	Extension to be added to the keyword if keyword is a file name. Default is ".asc"

`wpath` working directory containing GEOtop files (included the inpts file). It is mandatory if `raster` is `TRUE`. See `declared.geotop.inpts.keywords`.

`add_wpath` logical value. Default is `FALSE`. If `TRUE`, the `wpath` string is attached to the keyword string value. It is automatically set `TRUE` if `raster` is `TRUE`.

`use.read.raster.from.url`
logical value. Default is `TRUE`. If `TRUE` the RasterLayer are read with `read.raster.from.url`, istead of `raster` (otherwise). It is recomended in case the files whose paths are contained in `x` are remote and are 'http' addresses. In this cases the stand-alone method `raster(x)` does not always work and `use.read.raster.from.url` is necessary.

`data.frame` logical value. It is an option for tabular data. If `TRUE` function returns direclty a data frame or a list of data frames as `data.frame` or `zoo` objects imported from the keyword-related files using `read.table` function. In this case the argument `wpath` (see `declared.geotop.inpts.keywords`) is mandatory. Default is `FALSE`.

`formatter` string value. It is the decimal formatter contained in the file name and used in case the tabular data are referred at several points. Default is `"%04d"` . It is used in case `data.frame` is `TRUE`.

`level` integer values. Numbers incating all the identandification numbers of the files containing the requested data frames. Default is 1, correspondig to the decimal formatter `"0001"`. See examples.

`date_field` string value. Default is "Date", otherwise defined by the value of `HeaderDateDDMMYYYYhhmmMeteo` geotop keyword. It is used only if the argument `data.frame` is `TRUE`. If it is `NULL` or `NA` the function return a list of generic `data.frame` object(s), otherwise `link{zoo}` object(s). See the arguments `tz` and `format` for Date formatting.

`isNA` numeric value indicating NA in geotop ascii files. Default is -9999.00

`matlab.syntax`
logical value. Default is `FALSE`. If `TRUE` a vector is written in a string according to *.m file syntax. Warning: this synstax is not read by GEOtop.

`projfile` fileneme of the GEOtop projection file. Default is `geotop.proj`.

`start_date,end_date`
null objects or dates in `POSIXlt` format between which the variables are returned. It is enabled in case that `date_field` is not `NULL` or `NA` and `data.frame` is `TRUE`. Default is `NULL`.

`ContinuousRecovery`
integer value. Default is 0. It is used for tabular output data and is the number of times GEOtop simulation broke during its running and was re-launched with 'Contiuous Recovery' option.

`ContinuousRecoveryFormatter`
character string. Default is `'_crec%04d'`. It is used only for tabular output data and if `ContinuousRecovery` is equal or greater than 1.

`...` further arguments of `declared.geotop.inpts.keywords`

Value

the keyword value

Note

If `inpts.frame` is `NULL`, `inpts.frame` will be obtained by calling the function `declared.geotop.inpts.keywo` with `...` arguments.

Examples

```
library(geotopbricks)

#Simulation working path
wpath <- 'http://www.boussinesq.org/geotopbricks/simulations/panola13_run2xC_test3'
prefix <- get.geotop.inpts.keyword.value("SoilLiqWaterPressTensorFile",wpath=wpath)

slope <- get.geotop.inpts.keyword.value("SlopeMapFile",raster=TRUE,wpath=wpath)
bedrock_depth <- get.geotop.inpts.keyword.value("BedrockDepthMapFile",raster=TRUE,wpath=wpat

layers <- get.geotop.inpts.keyword.value("SoilLayerThicknesses",numeric=TRUE,wpath=wpath)
names(layers) <- paste("L",1:length(layers),sep="")

##### set van genuchten parameters to estimate water volume
theta_sat <- get.geotop.inpts.keyword.value("ThetaSat",numeric=TRUE,wpath=wpath)
theta_res <- get.geotop.inpts.keyword.value("ThetaRes",numeric=TRUE,wpath=wpath)
alphaVG <-  get.geotop.inpts.keyword.value("AlphaVanGenuchten",
numeric=TRUE,wpath=wpath) # expressed in mm^-1

nVG <-  get.geotop.inpts.keyword.value("NVanGenuchten",numeric=TRUE,wpath=wpath)


##### end set van genuchten parameters to estimate water volume


##### set meteo data

start <-  get.geotop.inpts.keyword.value("InitDateDDMMYYYYhhmm",date=TRUE,wpath=wpath,tz="A"
end <- get.geotop.inpts.keyword.value("EndDateDDMMYYYYhhmm",date=TRUE,wpath=wpath,tz="A")

nmeteo <- get.geotop.inpts.keyword.value("NumberOfMeteoStations",numeric=TRUE,wpath=wpath)
level <- 1:nmeteo

# Not Run: uncomment the following lines to calculate "meteo"
# meteo <- get.geotop.inpts.keyword.value("MeteoFile",wpath=wpath,data.frame=TRUE,
#         level=level,start_date=start,end_date=end)
#

##### end set meteo data
```

`get.geotop.recovery.state`

This function saves all spatially distrubuted information contained in the recovery folder into a comprehensive `list` *object.*

Description

This function saves all spatially distrubuted information contained in the recovery folder into a comprehensive `list` object.

Usage

```
get.geotop.recovery.state(recFolder, xx = "0000", formatter = "L%04d",
  extension = ".asc", nsoillayers = 10, ...)
```

Arguments

`recFolder`	directory when recvery maps are set. In GEOtop it is ...
`xx`	charcter String. Default is `"0000"`
`extension`	file estension used for asccii recovery map files. It must contains `'.'` as the first character. Defaut is `".asc"` .
`formatter`	string character for the the decimal formatter to be used. Default is `"L%04d"`.
`nsoillayers`	number of soil layers used in the GEOtop simulation
`...`	further arguments

Value

a `list` object containining all recovery raster maps.

Note

This function has been used with the built 1.225-9 of GEOtop .

Author(s)

Emanuele Cordano

See Also

`brick.decimal.formatter`,

`raster`,`set.geotop.recovery.state`,

`write.vectorized.geotop.recovery`,`read.vectorized.geotop.recovery`

Examples

```
library(geotopbricks)
example_Rscript <- system.file('template/example.geotop.recovery.state.R',package="geotopbri
example_Rscript

# Not Run because it elapses too long time!!!
# Please Uncomment the following line to run by yourself!!!
# source(example_Rscript)
```

`getProjection` *It reads the CRS metadata utilzed in a GEOtop Simulation*

Description

It reads the CRS metadata utilzed in a GEOtop Simulation

Usage

```
getProjection(x, cond = TRUE, ...)
```

Arguments

`x`	name and full path of the file containimg CRS information
`cond`	logical value. If `FALSE` the function returns `NA`. Default is `TRUE`.
`...`	futher arguments

Value

A string corresponding the projection and CRS if the argument `cond` is `TRUE`.

Examples

```
library(geotopbricks)
wpath <- "http://www.boussinesq.org/geotopbricks/simulations/idroclim_test1"
x <- paste(wpath,"geotop.proj",sep="/")


crs <- getProjection(x)
```

`getvalues.brick.at.depth`
Interpolates the values of a 'brick' at a certain depth and returns the map of brick values at the "depth" level

Description

Interpolates the values of a 'brick' at a certain depth and returns the map of brick values at the "depth" level

Usage

```
getvalues.brick.at.depth(x, depth, layers, i0 = NULL, verify = FALSE, ...)
```

Arguments

x	a 'RasterBrick' or a three-dimensional array
depth	depth map, generally a 'RasterLayer' object
layers	vector of layer thickness
i0	a 'Raster' containing the number of soil laver just over the bedrock. Default is NULL and is then calculated.
verify	logical. Default is FALSE. If it is TRUE, it verifies that function is working correctly.
...	further argument

Value

a list of 'Raster' maps:

i0 a 'Raster' containing the number of soil laver just over the bedrock

val_z0 a 'Raster' containing the values of x at the i0-th layer

val_z1 a 'Raster' containing the values of x at the (i0+1)-th layer

z0 a 'Raster' containing the depth of the center of the i0-th layer

z1 a 'Raster' containing the depth of the center of the (i0+1)-th layer

Note

x and depth or i0 must cover the same spatial region.

See Also

codevertical.aggregate.brick.within.depth

Examples

```
library(geotopbricks)
# The examples is the following R script conteined in a 'inst' directory of the package sour
f <- system.file("doc/examples/example.getvalues.brick.at.depth.R",package="geotopbricks")
#  source(f) # Uncomment this line to run the example.
# You can copy the example file using file.copy(from=f,to=....,...) See file.copy documentat
```

KML *KML*

Description

KML method for a GeotopRasterBrick object

Usage

```
## S4 method for signature 'GeotopRasterBrick'
KML(x, filename,
  crs = as.character("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"), ...)
```

Arguments

x	the GeotopRasterBrick object
filename	mane of the KML file to produce
crs	character string containg the LatLon reference system. Default is "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs" (see http://spatialreference.org/ref/epsg/4326/).
...	further argument for S4 method KLM for Raster object.

Note

A coordinate transformation is made with projectRaster.

Examples

```
library(geotopbricks)
# The examples is the following R script conteined in a 'inst' directory of the package sour
f <- system.file("doc/examples/example.KML.GeotopRasterBrick.R",package="geotopbricks")
#  source(f) # Uncomment this line to run the example.
# You can copy the example file using file.copy(from=f,to=....,...) See file.copy documentat
```

---

listFromOutputSoil3DTensor

Extracts a list of files pointing to an output 3D Tensor or 2D map respectively

---

Description

Extracts a list of files pointing to an output 3D Tensor or 2D map respectively

Usage

```
listFromOutputSoil3DTensor(x, when, layers = "SoilLayerThicknesses",
  one.layer = FALSE, suffix = "L%04dN%04d.asc", wpath = NULL,
  tz = "A", start_date_key = "InitDateDDMMYYYYhhmm",
  end_date_key = "EndDateDDMMYYYYhhmm", timestep = "OutputSoilMaps",
  use.read.raster.from.url = FALSE, crs = NULL, projfile = "geotop.proj",
  start.from.zero = FALSE, secondary.suffix = NULL, ...)
```

Arguments

x	string. GEOtop keyword reletated to the 3D or 2D variable to be imported in R.
when	POSIXlt-class for date and time on which the variable x is requested.
layers	number of soil layer or geotop keyword for soil leyer (e.g. SoilLayerThicknesses or SoilFile). Default is SoilLayerThicknesses.
timestep	time step expressed in seconds every which the raster file has been created. It can be a string corresponding to the geotop keyword in the inpts file. Default value is "OutputSoilMaps".
suffix	charcher string containing the decimal formatter used by GEOtop in the output file names. Default is "L not to modify the value of this argument and use the default value.
wpath,tz,use.read.raster.from.url	see get.geotop.inpts.keyword.value
projfile	name of the *.proj file containing CRS information. See get.geotop.inpts.keyword.value. Default is "geotop.proj". If is NULL or NA or this file does not exist, it is not searched and read.. In case use.read.raster.from.url is TRUE and no NULL or NA values are assinged, the *.proj file is searched.
crs,start.from.zero	see brick.decimal.formatter. If crs is not NULL (Default) , projfile is ignored.
one.layer	logical value. If TRUE a RasterLayer-class object is imported, otherwise a RasterBrick-classobject is returened. Default for brickFromOutputSoil3DTensor is FALSE
start_date_key,end_date_key	initial and final detes and times of the GEOtop simulation or alternatively the respective keywords of *.inpts file (Default)
secondary.suffix	String secondary suffix which can be added at the end of the Map file name (optional). Default is NULL and no secondary suffix is added.
...	additional arguments for get.geotop.inpts.keyword.value or brickFromOutputSoil3D

Details

This function is experimental and documentation partially exhaustive. These functions brickFromOutputSoil3DTenso and rasterFromOutput2DMap return 3D or 2D Raster-class objects respectively. rasterFromOutput2DMap is a wrapper function of brickFromOutputSoil3DTensor with the option one.layer==TRUE.
The functionswork with the following output keywords:

"SoilTempTensorFile",

"SoilAveragedTempTensorFile",

"SoilLiqContentTensorFile",

"SoilAveragedLiqContentTensorFile",

"SoilIceContentTensorFile",

"SoilAveragedIceContentTensorFile",

"SoilLiqWaterPressTensorFile",
"SoilTotWaterPressTensorFile" for `brickFromOutputSoil3DTensor`;
"FirstSoilLayerTempMapFile",
"FirstSoilLayerAveragedTempMapFile",
"FirstSoilLayerLiqContentMapFile",
"FirstSoilLayerIceContentMapFile",
"LandSurfaceWaterDepthMapFile",
"ChannelSurfaceWaterDepthMapFile",
"NetRadiationMapFile",
"InLongwaveRadiationMapFile",
"NetLongwaveRadiationMapFile",
"NetShortwaveRadiationMapFile",
"InShortwaveRadiationMapFile",
"DirectInShortwaveRadiationMapFile",
"ShadowFractionTimeMapFile",
"SurfaceHeatFluxMapFile",
"SurfaceSensibleHeatFluxMapFile",
"SurfaceLatentHeatFluxMapFile",
"SurfaceTempMapFile",
"PrecipitationMapFile",
"CanopyInterceptedWaterMapFile",
"SnowDepthMapFile",
"GlacierDepthMapFile",
"SnowMeltedMapFile",
"SnowSublMapFile",
"GlacierMeltedMapFile",
"GlacierSublimatedMapFile",
"AirTempMapFile",
"WindSpeedMapFile",
"WindDirMapFile",
"RelHumMapFile",
"SWEMapFile",
"GlacierWaterEqMapFile"
"SnowDurationMapFile",
"ThawedSoilDepthMapFile",
"ThawedSoilDepthFromAboveMapFile",
"WaterTableDepthMapFile",

"WaterTableDepthFromAboveMapFile",
"NetPrecipitationMapFile",
"EvapotranspirationFromSoilMapFile" for rasterFromOutput2DMap.

Author(s)

Emanuele Cordano

See Also

get.geotop.inpts.keyword.value,brick.decimal.formatter,brickFromOutputSoil3DTensor

Examples

```
tz <-  "Etc/GMT+1"
start <- as.POSIXct("2002-03-22",tz=tz)
end <- as.POSIXct("2002-03-25",tz=tz)
day <- 3600*24
when <- seq(from=start,to=end,by=day)
#' # The data containing in the link are only for educational use
wpath <- "http://www.boussinesq.org/geotopbricks/simulations/idroclim_test1"
x <- "SoilLiqContentTensorFile"
when <- as.POSIXct("2002-03-22 UTC",tz="A")

# Not Run because it elapses too long time!!!
# Please Uncomment the following lines to run by yourself!!!


# wpath <- '/Users/ecor/attivita/2013/fem-idroclima/Trentino_500_dstr_GEOtop_1_225_9_002'

#kpsi <- "SoilLiqWaterPressTensorFile" ## soil water pressure head

out <-listFromOutputSoil3DTensor(x,when=when,wpath=wpath,tz=tz,use.read.raster.from.url=FALS
```

---

max_value *max_value*

Description

Gets the maximum (scalar) values of a GeotopRasterBrick object

Usage

```
max_value(x)
```

Arguments

x	a GeotopRasterBrick object
...	further arguments

Value

the maximum (scalar) values of a GeotopRasterBrick object

---

min_value *min_value*

Description

Gets the minimum (scalar) values of a GeotopRasterBrick object

Usage

```
min_value(x)
```

Arguments

x	a GeotopRasterBrick object
...	further arguments

Value

the minimum (scalar) values of a GeotopRasterBrick object

Ops *Ops*

Description

Ops method for a GeotopRasterBrick object

Usage

```
## S4 method for signature 'GeotopRasterBrick,GeotopRasterBrick'
Ops(e1, e2)

## S4 method for signature 'GeotopRasterBrick,numeric'
Ops(e1, e2)

## S4 method for signature 'numeric,GeotopRasterBrick'
Ops(e1, e2)
```

Arguments

e1,e2	the GeotopRasterBrick or numeric objects

Note

If e1 or e2 time index is not taken into account.

plot *plot*

Description

plot method for a GeotopRasterBrick object

Usage

```
## S4 method for signature 'GeotopRasterBrick,ANY'
plot(x, y = NULL, ...)
```

Arguments

x	the GeotopRasterBrick object
y	further argument
...	further argument for S4 method plot for Raster object.

See Also

KML

Examples

```
library(geotopbricks)
# The examples is the following R script conteined in a 'inst' directory of the package sour
f <- system.file("doc/examples/example.plot.GeotopRasterBrick.R",package="geotopbricks")
#  source(f) # Uncomment this line to run the example.
# You can copy the example file using file.copy(from=f,to=....,...) See file.copy documentat
```

pointer.to.maps.xyz.time
pointer.to.maps.xyz.time

Description

pointer.to.maps.xy.time

Usage

```
pointer.to.maps.xyz.time(wpath, map.prefix = "thetaliq",
  suffix = "L%04dN%04d.asc", zoo.index = NULL, ntime, nlayers)
```

Arguments

wpath	complete working path to *.asc maps are saved
map.prefix	string prefix name map before
suffix	z-time or time suffix plus file extention character string. Default for GEOtop application is "L%04dN%04d.asc" for xy+z+time maps or "N%04d.asc" for xy+time maps.
zoo.index	time or date index. Default is NULL , otherwise function returns a zoo object with zoo.index as index.
ntime	number of time instant. If zoo.index is not NULL, it is calculated from zoo.index length.
nlayers	number of vertical layers.

Value

A dota.frame or zoo object containig the paths to maps fpr each time and z layer.

Author(s)

Emanuele Cordano

---

read.ascii.vectorized.brick

Read a text file containing values and matedata of a z-layer brick referred to a time instant (e.g. date). The file is formatted like an ascii format like 'geotop.inpts' file.

Description

Read a text file containing values and matedata of a z-layer brick referred to a time instant (e.g. date). The file is formatted like an ascii format like 'geotop.inpts' file.

Usage

```
read.ascii.vectorized.brick(file = NULL, comment = "!", crs = "",
  NAflag = -9999, matlab.syntax = FALSE, ...)
```

Arguments

file	file name to write
comment	character. Comment indicator. Default is "!".
NAflag	numeric. Dafauli is -9999, see writeRasterxGEOtop.
crs	Character or object of class CRS. PROJ4 type description of a Coordinate Reference System (map projection) (optional). See brick or raster.
matlab.syntax	logical value. Default is FALSE. If TRUE the file syntax is like the one of a *.m Matlab script file.
...	further aguments inserted as attribute

Value

the RasterBrick-class object

See Also

write.ascii.vectorized.brick

Examples

```
# see the examples of read.ascii.vectorized.brick
```

---

read.raster.from.url

It imports a 'RasterLayer' object in Escri-Asci format from a URL 'http://....<FILENAME>.asc

---

Description

It imports a 'RasterLayer' object in Escri-Asci format from a URL 'http://....<FILENAME>.asc

Usage

```
read.raster.from.url(x, header_nrow = 6, ...)
```

Arguments

x	the charcater string containing the URL address
header_nrow	Number of header in the ASCII grid format. Deafault is 6. See http://en.wikipedia.org/wiki/Esri_grid
...	additional arguments

Value

a 'RasterLayer' object

Note

This function reads a local or remote text files formatted as http://en.wikipedia.org/wiki/Esri_grid and creates a 'RasterLayer' object.

See Also

raster,readLines

`read.vectorized.geotop.recovery` *Reads a text file like the one generated by* `write.vectorized.geotop.recovery`

Description

#. containing values and matedata of a z-layer brick referred to a time instant (e.g. date). The file is formatted like an ascii format like `'geotop.inpts'` file.

Usage

```
read.vectorized.geotop.recovery(file = file, comment = "!",
  matlab.syntax = TRUE, xx = "0000", formatter = "L%04d",
  extension = ".asc", NAflag = -9999, crs = "", ...)
```

Arguments

`file`	file name to write
`comment`	character. Comment indicator. Default is `"!"`.
`formatter,extension,xx`	see `get.geotop.recovery.state`.
`NAflag`	numeric. Default is -9999, see `writeRasterxGEOtop`.
`crs`	Character or object of class CRS. PROJ4 type description of a Coordinate Reference System (map projection) (optional). See `brick` or `raster`.
`matlab.syntax`	logical value. Default is `TRUE`. If `TRUE` the file syntax is like the one of a *.m Matlab script file.
`...`	further aguments inserted as attribute

Value

a `list` object like `get.geotop.recovery.state`

See Also

`write.vectorized.geotop.recovery`

Examples

```
# see the examples of read.ascii.vectorized.brick
```

replace.keyword *It replaces some keyword values of* geotop.inpts *file with the ones of anoter* *.inpts *value*

Description

It replaces some keyword values of geotop.inpts file with the ones of anoter *.inpts value

Usage

```
replace.keyword(x, y = "geotop.inpts", file.output = NULL,
  write.file.output = TRUE, wpath = NULL, ...)
```

Arguments

x	filename of the *.inpts with the "new" keyword value
y	filename of the *.inpts with the "old" keyword value. Default is "geotop.inpts".
file.output	filename where to write the comprehensive new geotop.inpts file. If it is NULL (default), the fileneme is assigned by y.
write.file.output	logical value. If it is TRUE, the output of the function is written in he file file.output.
wpath	working path to the GEOtop simulation folder containing the x and y files.
...	further arguments

Details

This function repleces some keword values of y with the ones indicated in y. It is useful to replace the meteo station metedata, for instance, when the meteorological station of a study cases are modified. The function returns the new geotop.inpts file as a vector of character strings. If write.file.output==TRUE, the output is written in an extarnal file, e.g. "geotop.inpts" newly (this option is suggested).

Author(s)

Emanuele Cordano

Examples

```
library(geotopbricks)
wpath <- system.file('template/meteo_ex',package="geotopbricks")
x <- "meteo.inpts"
zl <- replace.keyword(x,wpath=wpath,write.file.output=FALSE)
```

set.geotop.recovery.state

This function re-writes the recovery ascii raster maps in a given folder

Description

This function re-writes the recovery ascii raster maps in a given folder

Usage

```
set.geotop.recovery.state(rec, newRecFolder, ...)
```

Arguments

`rec`	a `list` object returened by `get.geotop.recovery.state`
`newRecFolder`	directory where to write all recovery raster asccii maps
`...`	further arguments

Author(s)

Emanuele Cordano

See Also

`get.geotop.recovery.state`,`writeRasterxGEOtop`

Examples

```
# See the examples of the 'get.geotop.recovery.state' function
```

vertical.aggregate.brick.within.depth

Aggregates with a mean or an addition on the vertical profile the values of a 'brick' within a certain depth and returns the vertical aggregated map

Description

Aggregates with a mean or an addition on the vertical profile the values of a 'brick' within a certain depth and returns the vertical aggregated map

Usage

```
vertical.aggregate.brick.within.depth(x, depth = NULL, layers = NULL,
  i0 = NULL, verify = FALSE, FUN = identity, divide.by.depth = FALSE,
  ...)
```

Arguments

x	a 'RasterBrick' or a three-dimensional array
depth	depth map, generally a 'RasterLayer' object
layers	vector of layer thickness
i0	a 'Raster' containing the number of soil laver just over the bedrock. Default is NULL and is then calculated.
verify	logical. Default is FALSE. If it is TRUE, it verifies that function is working correctly.
FUN	function used for aggregation. If missing, identity is the default value.
divide.by.depth	logical. If TRUE the function returns the 'mean' value, otherwise a a cumulate value. Default is FALSE.
...	further argument for FUN

Value

a list of 'Raster' maps:

i0 a 'Raster' containing the number of soil laver just over the bedrock

z0 a 'Raster' containing the depth of the center of the i0-th layer

result a 'Raster' containing the aggregated map

Note

x and depth or i0 must cover the same spatial region.

See Also

getvalues.brick.at.depth,brick

Examples

```
library(geotopbricks)
# The examples is the following R script conteined
# in a 'inst' directory of the package source
f <- system.file("doc/examples/example.vertical.aggregate.brick.within.depth.R",
package="geotopbricks")
#  source(f) # Uncomment this line to run the example.
# You can copy the example file using file.copy(from=f,to=....,...) See file.copy documentat
```

`write.ascii.vectorized.brick` *Writes a z-layer brick referred to a time instant (e.g. date) in an ascii format like* `'geotop.inpts'` *file.*

Description

Writes a z-layer brick referred to a time instant (e.g. date) in an ascii format like `'geotop.inpts'` file.

Usage

```
write.ascii.vectorized.brick(b, file = NULL, header = NULL,
  overwrite = TRUE, NAflag = -9999, matlab.syntax = FALSE, ...)
```

Arguments

`b`	a `RasterBrick-class` or `GeotopRasterBrick-class` object
`file`	file name to write
`header`	character string vector for header text lines. If missing, a default header is written. #Default is `c("! header")`.
`overwrite`	logical. Default is `TRUE`, see `writeRaster`.
`NAflag`	numeric. Default is -9999, see `writeRasterxGEOtop`.
`matlab.syntax`	logical value. Default is `FALSE`. If `TRUE` the file syntax is like the one of a *.m Matlab script file.
`...`	further aguments inserted as attribute

Value

the string vector possibly written in `file`.

Note

Add Quote if necessary. This function is NOT mantained and will be DEPRECATED.

See Also

`read.ascii.vectorized.brick`

Examples

```
## Not Run
## library(geotopbricks)
## library(raster)
## file <- system.file("doc/examples/snowthickness",package="geotopbricks")
## file <- paste(file,"SnowThickness0000L%04d.asc",sep="/")
## b <- brick.decimal.formatter(file=file,nlayers=15)
## nlayers(b)
## names(b)
## file <- "snow.txt"
## btext <- write.ascii.vectorized.brick(b,Date="1/1/2009",file="snow.txt")
## The printed object
## str(btext)
## bb <- read.ascii.vectorized.brick(file = file)
## bf <- abs(as.matrix(bb[[1]]-b[[1]]))<.Machine$double.eps^0.5
```

`write.vectorized.geotop.recovery`

It writes a `list` *object returened by* `get.geotop.recovery.state` *as a string vector or in a text file, following* `*.inpts` *or Matlab-like syntax.*

Description

It writes a `list` object returened by `get.geotop.recovery.state` as a string vector or in a text file, following `*.inpts` or Matlab-like syntax.

Usage

```
write.vectorized.geotop.recovery(rec, file = NULL, header = NULL,
  overwrite = TRUE, NAflag = -9999, matlab.syntax = TRUE, ...)
```

Arguments

`rec`	a `list` object returened by `get.geotop.recovery.state`
`file`	ascii text file name whrere to write the string vector
`header`	character string vector for header text lines. If missing, a default header is written. Default is `c("! header")` or he one assigned by `matlab.syntax`.
`overwrite`	logical. Default is `TRUE`, see `writeRaster`.
`NAflag`	numeric. Default is -9999, see `writeRasterxGEOtop`.
`matlab.syntax`	logical value. Default is `TRUE`. If `TRUE` the file syntax is like the one of a *.m Matlab script file.
`...`	further aguments inserted as attribute

Value

a string vector containg the `rec` variables.

Note

Add Quote if necessary

See Also

`get.geotop.recovery.state`,`set.geotop.recovery.state`,`write.vectorized.variable.in.str`

Examples

```
# See the examples of the 'get.geotop.recovery.state' function
```

`write.vectorized.variable.in.string`

Writes one or more variables (scalars, vectors or Rasters) in a string each, following `*.inpts` or Matlab-like syntax.

Description

Writes one or more variables (scalars, vectors or Rasters) in a string each, following `*.inpts` or Matlab-like syntax.

Usage

```
write.vectorized.variable.in.string(l, NAflag = -9999,
  matlab.syntax = FALSE, ...)
```

Arguments

`l`	a codelist object contained the variables (scalars, vectors or Rasters) which will be written in a string each.
`NAflag`	numeric. Default is -9999, see `writeRasterxGEOtop`.
`matlab.syntax`	logical value. Default is `FALSE`. If `TRUE` the file syntax is like the one of a *.m Matlab script file.
`...`	further aguments

Value

the string vector `<NAME_VARIABLE>==<VALUES_VARIABLE>`.

Note

Add Quote if necessary

See Also

`read.ascii.vectorized.brick`

Examples

```
a <- 1:5
l <- list(v=a,a=a)
out <- write.vectorized.variable.in.string(l,matlab.syntax=TRUE)
out
```

---

writeRasterxGEOtop *This function uses* `writeRaster` *to create .asc maps which can be read by GEOtop*

Description

This function uses `writeRaster` to create .asc maps which can be read by GEOtop

Usage

```
writeRasterxGEOtop(x, filename = NULL, overwrite = TRUE, NAflag = -9999,
  use.decimal.formatter = FALSE, start.from.zero = FALSE, keyword, wpath,
  suffix.ext = ".asc", ...)
```

Arguments

`x`	a Raster object, see `writeRaster`. It can be also a `RasterBrick-class` object.
`filename`	see `writeRaster`. It is a vector of string or one string containing a decimal formatter (see `brick.decimal.formatter`) in case `x` is a `RasterBrick-class` object.
`overwrite`	logical. Default is `TRUE`, see `writeRaster`.
`NAflag`	numeric. Dafauli is -9999, see `writeRaster`.
`use.decimal.formatter`	logical value. Default is `FALSE`. If it is `TRUE` or `x` is a `RasterBrick-class` object with `nlayers(x)!=length(filename)` , `filename` is considered as one string containing a decimal formatter (e.g. `"%04d"`, see `brick.decimal.formatter`). Otherwise, if `filename` is considered as a vector string.
`start.from.zero`	logical value. Default is `FALSE`. If `TRUE` the formatter starts from `0000`, otherwise it starts from `0001`.
`keyword`	geotop keyword to be used to extract the raster file name from `geotop.inpts` file. This is enabled if `filename` is equal to `NULL`.
`wpath`	simulation folder containing `geotop.inpts` file.
`suffix.ext`	charachter string to be added to the `keyword` value,e.g. possible suffix and extension of the raster file name. Default is `".asc"`.
`...`	further arguments of `get.geotop.inpts.keyword.value` or `writeRaster`

Note

It makes use of `system` functons. It uses `*.asc` format for raster files. In case the file name `filename` is missing and then `NULL`, it must be imported by the simulation `geotop.inpts` file.

`zoo-class` *A GeotopRasterBrick: an object to manage raster maps provied by GEOtop!!*

Description

A GeotopRasterBrick: an object to manage raster maps provied by GEOtop!!

Examples

```
showClass("zoo")
```