

Package ‘RFimex’

June 2, 2014

Type Package

Title R interface for Fimex

Version 0.04

Date 2014-02-11

Author Heiko Klein

Maintainer Heiko Klein <Heiko.Klein@met.no>

Description R interface for Fimex, a library to read and manipulate gridded data

License LGPL (>= 2.0, < 3)

R topics documented:

| | |
|--|-----------|
| RFimex-package | 2 |
| mifi.reader.getCoordinates | 3 |
| mifi.reader.getProj4 | 5 |
| mifi.reader.getSliceVecInUnit | 6 |
| mifi.reader.lonLatInterpolated | 7 |
| mifi.reader.new | 8 |
| mifi.reader.uniqueRefTime | 9 |
| mifi.reader.variables | 10 |
| mifi.reader.vectorAutoRotated | 11 |
| mifi.reader.write | 12 |
| mifi.sb.getDimensions | 13 |
| mifi.sb.new | 14 |
| mifi.sb.setStartAndSize | 15 |
| Index | 17 |

RFimex-package

Access to gridded data via Fimex

Description

The RFimex package exposes some of the Fimex library functionality to access gridded data, eventually interpolated gridded data.

Details

Package: RFimex
Type: Package
Version: 0.04
Date: 2014-02-11
License: LGPL-2

Author(s)

Heiko Klein Maintainer: Heiko Klein <heiko.klein@met.no>

References

<http://fimex.met.no>

See Also

[mifi.reader.new](#), [mifi.reader.lonLatInterpolated](#), [mifi.reader.vectorAutoRotated](#), [mifi.reader.variables](#), [mifi.reader.getCoordinates](#), [mifi.reader.getProj4](#), [mifi.reader.getSliceVecInUnit](#), [mifi.reader.write](#), [mifi.sb.new](#), [mifi.sb.getDimensions](#), [mifi.sb.getStartAndSize](#),

Examples

```
library(RFimex)
# real world reader
reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}
#reader <- mifi.reader.new("netcdf", "out.nc", "")
#reader <- mifi.reader.new("netcdf", "http://thredds.met.no/thredds/dodsC/metno/proff4km/default/Proff_Default_

vars <- mifi.reader.variables(reader)
# check if "altitude" is part of vars
# ...
```

```

altCoords <- mifi.reader.getCoordinates(reader, "altitude")
sb <- mifi.sb.new(reader, "altitude")
if (!is.na(altCoords["time"])) {
  # just fetch first time
  mifi.sb.setStartAndSize(sb, altCoords["time"], 0, 1)
}
# just fetch on corner of size 3 * 5
if (!is.na(altCoords["x"])) {
  mifi.sb.setStartAndSize(sb, altCoords["x"], 0, 3)
}
if (!is.na(altCoords["y"])) {
  mifi.sb.setStartAndSize(sb, altCoords["y"], 0, 5)
}
# altitude data in cm
altData <- mifi.reader.getSliceVecInUnit(reader, "altitude", sb, "cm")
if (length(altData) != 15) {
  stop("didn't get all data; got ", length(altData))
}

altProj4 <- mifi.reader.getProj4(reader, "altitude")
if (altProj4 == "") {
  stop("unable to read proj4-string for altitude; got ", altProj4)
}

```

```

mifi.reader.getCoordinates
      mifi.reader.getCoordinates-internals

```

Description

Get the identified coordinate-dimensions from a variable.

Usage

```
mifi.reader.getCoordinates(reader, varName)
```

Arguments

| | |
|---------|---|
| reader | a reader as retrieved from mifi.reader.new |
| varName | a variable name as retrieved from mifi.reader.variables |

Details

This will try to identify the coordinate-system with the variable, and return all dimensions.

Value

| | |
|---------|---|
| time | Name of the time-dimension |
| x | Name of the x-dimension (or longitude) |
| y | Name of the y-dimension (or latitude) |
| z | Name of the vertical-dimension (height, sigma, ...) |
| refTime | Name of the reference-time dimension |
| other1 | Name of the first unknown dimensions |
| other2 | Name of the 2. unknown dimensions |
| other3 | Name of the 3. unknown dimensions |
| other4 | Name of the 4. unknown dimensions |
| ... | |

Warning

If no dimensions are return, this does not mean, that the variable is dimension-less. It just means, that no coordinate-system could be found.

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#), [mifi.reader.variables](#), [mifi.sb.getDimensions](#)

Examples

```
# R example for RFimex
library(RFimex)

reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}
coords <- mifi.reader.getCoordinates(reader, "altitude")
# coords = c(x = x, y = y)
```

`mifi.reader.getProj4` *mifi.reader.getProj4-internals*

Description

Get the identified projection as proj4-string for a variable.

Usage

```
mifi.reader.getProj4(reader, varName)
```

Arguments

| | |
|---------|--|
| reader | a reader as retrieved from <code>mifi.reader.new</code> |
| varName | a variable name as retrieved from <code>mifi.reader.variables</code> |

Details

This will try to identify the coordinate-system with the variable, and return the projection as proj4-string

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#), [mifi.reader.variables](#), [mifi.sb.getDimensions](#)

Examples

```
# R example for RFimex
library(RFimex)

reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}
proj4 <- mifi.reader.getProj4(reader, "altitude")
# proj4 = "+proj=latlon +R=3671000"
```

```
mifi.reader.getSliceVecInUnit
      mifi.reader.getSliceVecInUnit-internals
```

Description

Fetch a data, or a data-slice from the data-source.

Usage

```
mifi.reader.getSliceVecInUnit(reader, varName, sb, units)
```

Arguments

| | |
|---------|--|
| reader | a reader, e.g. from <code>mifi.reader.new</code> |
| varName | a variable name |
| sb | a slice-builder, belonging to the variable name, eventually reduced by <code>mifi.sb.setStartAndSize</code> |
| units | units to fetch the data from. units will be automatically converted. If the original data does not have units, fetching will fail. |

Value

A data-array.

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#), [mifi.sb.new](#)

Examples

```
# R example for RFimex
library(RFimex)

reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}

sb <- mifi.sb.new(reader, "time")
mifi.sb.setStartAndSize(sb, "time", 3, 5)

time <- mifi.reader.getSliceVecInUnit(reader, "time", sb, "seconds since 2007-05-16 00:00:00 +0000")
# time has now size 5 and starts from the 4th position in the file
```

mifi.reader.lonLatInterpolated
mifi.reader.lonLatInterpolated-internals

Description

Read data interpolated to a list of latitude longitude points from a datasource

Usage

```
mifi.reader.lonLatInterpolated(reader, method, lons, lats)
```

Arguments

| | |
|--------|--|
| reader | a CDMReader as created by mifi.reader.new |
| method | interpolation method, see http://fimex.met.no/doc/mifi__constants_8h_source.html MIFI_INTERPOL_* constants, e.g. NEAREST_NEIGHBOR, BILINEAR, BICUBIC, COORD_NN, COORD_NN_KD, FORWARD_SUM, FORWARD_MEAN, FORWARD_MEDIAN, FORWARD_MAX, FORWARD_MIN |
| lons | vector of longitude values in degree |
| lats | vector of latitude values in degree, must be same length as lons |

Details

The data-fetching is only prepared in this step. Data is interpolated during data-fetching, e.g. mfi.reader.getSliceVecInUnit

Value

a CDMReader object

Note

Vector properties, like wind or current are rotated if the spatial_direction property is set in the setup file, or if the standard_name is known, e.g. x_wind, *_x_velocity, *_x_transport, *_x_displacement. The complete list can be seen in https://svn.met.no/viewvc/fimex/trunk/src/coordSys/CF1_xCoordSysBuilder.cc?view=markup (enhanceVectorProperties)

The output will be of size: x=length(lons), y=1

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#)

Examples

```
library(RFimex)

reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}
lats <- c(70:90);
lons <- c(-10:10);
iread <- mifi.reader.lonLatInterpolated(reader, 1, lons, lats);
```

| | |
|-----------------|----------------------------------|
| mifi.reader.new | <i>mifi.reader.new-internals</i> |
|-----------------|----------------------------------|

Description

create a new reader object, e.g. a file-handle

Usage

```
mifi.reader.new(type, filename, config)
```

Arguments

| | |
|----------|--|
| type | file-type, e.g. felt, netcdf, grib |
| filename | source of data, e.g. filename or file-URL (in the case of opendap) |
| config | configuration of dataFile, required for felt and grib-files, optional for netcdf |

Value

Returns a CDMReader object. This function might fail, use try

Author(s)

Heiko Klein

Examples

```
library(RFimex)

## The function is currently defined as
reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}

## examples with netcdf or opendap (without "try")
```



```
#reader <- mifi.reader.new("netcdf", "out.nc", "")  
#reader <- mifi.reader.new("netcdf", "http://thredds.met.no/thredds/dodsC/metno/proff4km/default/Proff_Default_
```

`mifi.reader.uniqueRefTime`

mifi.reader.uniqueRefTime-internals

Description

fetch a vector of available uniqueRefTime from a reader

Usage

```
mifi.reader.uniqueRefTime(reader, units)
```

Arguments

| | |
|--------|--|
| reader | a reader as retrieved from <code>mifi.reader.new</code> , or a derivated reader like <code>mifi.reader.lonLatInterpolated</code> |
| units | a time unit, like "seconds since 1970-01-01 00:00:00 +0000" |

Details

Lists all, also auxiliary variable-names of a source.

Value

the unique reference time of the file, if only one available

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#)

Examples

```
library(RFimex)  
  
## The function is currently defined as  
reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))  
if (inherits(reader, "try-error")) {  
  stop("mifi.reader.new failed")  
}  
  
refTime <- mifi.reader.uniqueRefTime(reader, "seconds since 1970-01-01 00:00:00 +0000")
```

mifi.reader.variables *mifi.reader.variables-internals*

Description

fetch a vector of available variables from a reader

Usage

```
mifi.reader.variables(reader)
```

Arguments

reader a reader as retrieved from `mifi.reader.new`, or a derivated reader like `mifi.reader.lonLatInterpolated`

Details

Lists all, also auxiliary variable-names of a source.

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#), [mifi.reader.lonLatInterpolated](#)

Examples

```
library(RFimex)

## The function is currently defined as
reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}

vars <- mifi.reader.variables(reader)
```

mifi.reader.vectorAutoRotated
mifi.reader.vectorAutoRotated-internals

Description

Read data in original grid, but with vectors rotated to the grid, or to latlon

Usage

```
mifi.reader.vectorAutoRotated(reader, toLatLon)
```

Arguments

| | |
|----------|--|
| reader | a CDMReader as created by mifi.reader.new |
| toLatLon | a value of 0 will rotate the lat/lon vectors to the grid, a value of 1 will rotate the x/y-vectors to north/east |

Details

The data-fetching is only prepared in this step. Vectors are rotated during the fetch, e.g. mifi.reader.getSliceVecInUnit

Value

a CDMReader object

Note

Vector properties, like wind or current are rotated if the spatial_direction property is set in the setup file, or if the standard_name is known, e.g. x_wind, *_x_velocity, *_x_transport, *_x_displacement. The complete list can be seen in https://svn.met.no/viewvc/fimex/trunk/src/coordSys/CF1_xCoordSysBuilder.cc?view=markup&enhanceVectorProperties

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#)

Examples

```
library(RFimex)

reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}
iread <- mifi.reader.vectorAutoRotated(reader, 1);
```

mifi.reader.write *mifi.reader.write-internals*

Description

Write a reader to a file, currently only netcdf-files implemented.

Usage

```
mifi.reader.write(reader, type, filename, configname = "")
```

Arguments

| | |
|------------|--|
| reader | a reader as retrieved from <code>mifi.reader.new</code> , or a derivated reader like <code>mifi.reader.lonLatInterpolated</code> |
| type | filetype, must be "netcdf", "netcdf3" or "netcdf4" currently |
| filename | filename of the created file |
| configname | optional netcdf-config file |

Details

returns nothing

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#), [mifi.reader.lonLatInterpolated](#)

Examples

```
library(RFimex)

## The function is currently defined as
reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}

status <- try(mifi.reader.write(reader, "netcdf", "outTest.nc"))
if (inherits(status, "try-error")) {
  stop("mifi.reader.write failed")
}
```

mifi.sb.getDimensions *mifi.sb.getDimensions-internals*

Description

List all dimensions used in a SliceBuilder

Usage

```
mifi.sb.getDimensions(sb)
```

Arguments

sb SliceBuilder as retrieved from mifi.sb.new

Value

List of all dimensions of the variable given to the SliceBuilder

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#), [mifi.sb.new](#)

Examples

```
# R example for RFimex
library(RFimex)

reader <- try(mifi.reader.new("felt", "../test/flth00.dat", "../share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}

sb <- mifi.sb.new(reader, "time")
dims <- mifi.sb.getDimensions(sb)
# dims contains now just "time" dimension
```

mifi.sb.new

mifi.sb.new-internals

Description

Prepare an SliceBuilder for fetching a data-slice from a variable.

Usage

```
mifi.sb.new(reader, varName)
```

Arguments

| | |
|---------|---|
| reader | a reader, e.g. fetched from mifi.reader.new |
| varName | a variable name, e.g. one of those given in mifi.reader.variables |

Value

Returns a slicebuilder object.

Author(s)

Heiko Klein

See Also

[mifi.reader.new](#), [mifi.reader.variables](#)

Examples

```
# R example for RFimex
library(RFimex)

reader <- try(mifi.reader.new("felt", "../..test/flth00.dat", "../..share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
  stop("mifi.reader.new failed")
}

sb <- mifi.sb.new(reader, "time")
```

mifi.sb.setStartAndSize

mifi.sb.setStartAndSize-internals

Description

Prepare data-fetching to slice data along a dimension.

Usage

```
mifi.sb.setStartAndSize(sb, dimName, start, size)
```

Arguments

| | |
|---------|--|
| sb | a slicebuilder as retrieved from mifi.sb.new |
| dimName | dimension name of the variable of the slicebuilder |
| start | start-position of the slice, starting at 0 |
| size | size of the new slice |

Author(s)

Heiko Klein

See Also

[mifi.sb.new](#), [mifi.reader.getSliceVecInUnit](#)

Examples

```
# R example for RFimex
library(RFimex)

reader <- try(mifi.reader.new("felt", "../..test/flth00.dat", "../..share/etc/felt2nc_variables.xml"))
if (inherits(reader, "try-error")) {
```

```
        stop("mifi.reader.new failed")
    }
    sb <- mifi.sb.new(reader,"time")
    mifi.sb.setStartAndSize(sb,"time", 3, 5)

    time <- mifi.reader.getSliceVecInUnit(reader, "time", sb, "seconds since 2007-05-16 00:00:00 +0000")
    # time has now size 5 and starts from the 4th position in the file
```


Index

*Topic **\textasciitildekwd1**

`mifi.reader.new`, 8

*Topic **\textasciitildekwd2**

`mifi.reader.new`, 8

*Topic **package**

RFimex-package, 2

`mifi.reader.getCoordinates`, 2, 3

`mifi.reader.getProj4`, 2, 5

`mifi.reader.getSliceVecInUnit`, 2, 6, 15

`mifi.reader.lonLatInterpolated`, 2, 7, 10,
12

`mifi.reader.new`, 2, 4–7, 8, 9–14

`mifi.reader.uniqueRefTime`, 9

`mifi.reader.variables`, 2, 4, 5, 10, 14

`mifi.reader.vectorAutoRotated`, 2, 11

`mifi.reader.write`, 2, 12

`mifi.sb.getDimensions`, 2, 4, 5, 13

`mifi.sb.getStartAndSize`, 2

`mifi.sb.new`, 2, 6, 13, 14, 15

`mifi.sb.setStartAndSize`, 15

RFimex (RFimex-package), 2

RFimex-package, 2