Package: gdx2 (via r-universe)

August 31, 2024

Type Package

Title Interface package for GDX files in R

Version 0.3.2

Date 2024-08-01

Description A wrapper package for the gamstransfer package extending its functionality and allowing to read GDX files directly in R. It is emulating the basic features of the readGDX function in the gdx package but now based on gamstransfer instead of gdxrrw which served as a basis for gdx.

License BSD_2_clause + file LICENSE

URL `https://github.com/pik-piam/gdx2`

BugReports `https://github.com/pik-piam/gdx2/issues`

Imports gamstransfer, magclass (>= 6.0)

Suggests covr

Encoding UTF-8

LazyData yes

RoxygenNote 7.3.2

Config/Keywords tool

Repository https://pik-piam.r-universe.dev

RemoteUrl https://github.com/pik-piam/gdx2

RemoteRef HEAD

RemoteSha 50953b84e943320703ce313de42c8a568b16d70b

Contents

calcScaling 2
readGDX 3

Index **5**

calcScaling *calcScaling*

Description

This function creates a GAMS file with scaling of variables. The scaling is calculated based on a gdx file containing all variables of a run.

Usage

```
calcScaling(gdx, file = NULL, magnitude = 2)
```

Arguments

`gdx`	path to a gdx file
`file`	A file name the scaling GAMS code should be written to. If NULL the code is returned by the function
`magnitude`	The order of magnitude for which variables should be scaled. All variables with average absolute values which are either below 10e(-magnitude) or above 10e(magnitude) will be scaled.

Value

A vector with the scaling GAMS code if file=NULL, otherwise nothing is returned.

Author(s)

Jan Philipp Dietrich

See Also

`readGDX`

Examples

```
## Not run:
calcScaling("fulldata.gdx")

## End(Not run)
```

readGDX *readGDX*

Description

Function to read gdx files in R. It is a stripped-down reimplementation of readGDX which is now based on magclass structures and uses `gamstransfer` as basis.

Usage

```
readGDX(
  gdx,
  ...,
  format = "simplest",
  type = NULL,
  react = "warning",
  followAlias = FALSE,
  spatial = NULL,
  temporal = NULL,
  magpieCells = TRUE,
  select = NULL,
  restoreZeros = TRUE,
  addAttributes = TRUE
)
```

Arguments

`gdx` file name of a gdx file

`...` search strings defining the objects that should be read from gdx file, with *-autocompletion. Can also be vectors containing more than one search strings

`format` Output format. Five choices are currently available `simple`, `simplest`, `first_found`, and `name`. Instead of writing the full format name each format has its own abbreviation as shown below.

simple (s) This returns a list of outputs.

simplest (st) Behaves like "simple" if more than one object is returned. However, if only one object is read from gdx file the magpie object itself is returned getting rid of the surrounding list structure. This is the recommended format for interactive use.

first_found (f) This is a special format for the case that you would like to read in exactly one object but you do not know exactly what the name of the object is. Here, you can list all possible names of the object and the function will return the first object of the list which is found. This is especially useful writing read functions for gdx outputs of models in which the names of a data object might change over time but the function itself should work for all model versions. Having this format helps to make your gdx-based functions backwards compatible to older versions of a gdx file with different naming.

- **raw (r)** This returns the data as it comes from `gamstransfer::readGDX` without data class conversion.
- **name (n)** In this case the function returns the name of all objects found in the gdx which fit to the given search pattern and the given type as vector.

`type` Type of objects that should be extracted. Available options are "Parameter", "Set", "Alias", "Variable" and "Equation". If NULL all types will be considered.

`react` determines the reaction, when the object you would like to read in does not exist. Available options are "warning" (NULL is returned and a warning is send that the object is missing), "silent" (NULL is returned, but no warning is given) and "error" (The function throws out an error)

`followAlias` bolean deciding whether the alias or its linked set should be returned.

`spatial` argument to determine the spatial columns in the dataframe to be converted to a magclass object. Defaults to NULL. See `as.magpie` for more information.

`temporal` argument to determine the temporal columns in the dataframe to be converted to a magclass object. Defaults to NULL. See `as.magpie` for more information.

`magpieCells` (boolean) determines whether a set "j" gets special treatment by replacing underscores in the set elements with dots. Active by default for historical reasons. Can be ignored in most cases. Makes only a difference, if 1) GDX element depends on set "j", 2) set "j" contains underscores.

`select` preselection of subsets in the data coming from the gdx using the function `mselect`. Information has to be provided as a list of selections (e.g. `select=list(type="level")`). See `mselect` for more information.

`restoreZeros` Defines whether 0s, which are typically not stored in a gdx file, should be restored or ignored in the output. By default they will be restored. If possible, it is recommended to use restore_zeros=TRUE. It is faster but more memory consuming. If you get memory errors you should use restore_zeros=FALSE

`addAttributes` Boolean which controls whether the description and gdxMetadata should be added as attributes or not

Value

The gdx objects read in the format set with the argument `format`.

Author(s)

Jan Philipp Dietrich

Examples

```
## Not run:
readGDX("bla.gdx", "blub*")

## End(Not run)
```

Index

as.magpie, *4*

calcScaling, 2

mselect, *4*

readGDX, *2*, 3