Package: gdxdt (via r-universe)

October 21, 2024

Title IO for GAMS GDX Files using data.table

Version 0.1.0

Author Alois Dirnaichner [aut, cre]

Maintainer Alois Dirnaichner <alodi@directbox.com>

Description Interfaces GAMS data (*.gdx) files with data.tables using the GAMS R package gdxrrw. The gdxrrw package is available on the GAMS wiki: <https://support.gams.com/doku.php?id=gdxrrw:interfacing_gams_and_r>.

Depends R (>= 3.1), data.table (>= 1.11.0),

License MIT + file LICENCE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests gdxrrw, testthat

Repository https://loisel.r-universe.dev

RemoteUrl https://github.com/loisel/gdxdt

RemoteRef HEAD

RemoteSha 8a6a9e964bfa5aa722ca2b48d30d37eb9f700825

Contents

raw2dt *raw2dt*

Description

Provided the raw output from gdxrrw::rgdx, create a data.table with the correct UEL dimensions.

Usage

```
raw2dt(full_data)
```

Arguments

full_data	a list as given by gdxrrw::rgdx.

Value

a data.table

Examples

```
## Not run:
# prepare raw data
dt <- as.data.table(mtcars, keep.rownames = TRUE)
writegdx("test.gdx", dt, "test_var", valcol="wt", uelcols="rn", type="parameter")
data <- gdxrrw::rgdx("test.gdx", list(name="test_var"))
# data.table from raw data
table <- raw2dt(data)

## End(Not run)
```

raw2gdx *raw2gdx*

Description

Save to a GAMS gdx file. Works on a named list providing domains and data as given by gdxrrw::rgdx. This is a *workaround* to fix bugs in the implementation of gdxrrw::wgdx, namely the problems that domains are lost when writing the output of gdxrrw:rgdx and that for variables, a '_field' domain has always to be given. Using this wrapper, round-tripping data between R and gdx files should be possible.

Usage

```
raw2gdx(gdx, var)
```

Arguments

gdx	the gdx filename.
var	list of properties of a gdx symbol as provided by gdxrrw::rgdx.

Examples

```
## Not run:
# prepare raw data
dt <- as.data.table(mtcars, keep.rownames = TRUE)
writegdx("test.gdx", dt, "test_var", valcol="wt", uelcols="rn", type="parameter")
# round-trip
raw2gdx("test.gdx", gdxrrw::rgdx("test.gdx", list(name="test_var"))

## End(Not run)
```

---

readgdx *readgdx*

Description

Read a variable, parameter or set from a gdx file to a data.table.

Usage

```
readgdx(fname, varname, field = NULL)
```

Arguments

fname	the gdx filename.
varname	name of the object to load.
field	(for variable), select a field (default="l").

Value

a data.table

Examples

```
## Not run:
dt <- as.data.table(mtcars, keep.rownames = TRUE)
writegdx("test.gdx", dt, test_var, valcol="wt", uelcols="rn", type="parameter")
new_dt <- readgdx("test.gdx", test_var, type="parameter")

## End(Not run)
```

writegdx *writegdx*

Description

Save a data.table to a GAMS gdx file.

Usage

```
writegdx(gdx, dt, name, valcol, uelcols, type = "parameter",
  field = "l")
```

Arguments

gdx	the gdx filename.
dt	a data.table.
name	name of the variable.
valcol	name of data column.
uelcols	vector of column names with index dimensions.
type	type of symbol (variable or parameter)
field	the field if 'type == 'variable''

Examples

```
## Not run:
dt <- as.data.table(mtcars, keep.rownames = TRUE)
writegdx("test.gdx", dt, "test_var", valcol="wt", uelcols="rn", type="parameter")

## End(Not run)
```

writegdx.parameter *writegdx.parameter*

Description

Save a data.table to a parameter in a GAMS gdx file.

Usage

```
writegdx.parameter(gdx, dt, name, valcol, uelcols)
```

Arguments

gdx	the gdx filename.
dt	a data.table.
name	name of the parameter.
valcol	name of data column.
uelcols	vector of column names with index dimensions.

Examples

```
## Not run:
dt <- as.data.table(mtcars, keep.rownames = TRUE)
writegdx.parameter(test_gdx, dt, test_var, valcol="wt", uelcols="rn")

## End(Not run)
```

---

writegdx.variable *writegdx.variable*

Description

Save a data.table to a variable in a GAMS gdx file.

Usage

```
writegdx.variable(gdx, dt, name, valcol, uelcols, field = "l")
```

Arguments

gdx	the gdx filename.
dt	a data.table.
name	name of the variable.
valcol	name of data column.
uelcols	vector of column names with index dimensions.
field	the field if 'type == 'variable''

Examples

```
## Not run:
dt <- as.data.table(mtcars, keep.rownames = TRUE)
writegdx.variable(test_gdx, dt, test_var, valcol="wt", uelcols="rn")

## End(Not run)
```

Index