

Package: gdxrrw (via r-universe)

September 26, 2024

Title An Interface Between 'GAMS' and R

Version 1.0.10.9001

Depends R (>= 3.0)

Date 2021-04-02

SystemRequirements GAMS (>= 33.0.0)

Description A data interface between 'GAMS' and R. The 'GAMS' (General Algebraic Modeling System) software includes a data specification called 'GDX' that is the preferred way to store and exchange 'GAMS' data. This package includes several functions to transfer data between 'GDX' and R, and some related utility functions.

URL <http://www.gams.com>

License EPL2 with Secondary License GPL-2.0 or greater

Imports reshape2

LazyData no

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

RemoteUrl <https://github.com/pascal-sauer/gdxrrw>

RemoteRef HEAD

RemoteSha 4db6a9ded45cf23c063a09850b90004fce386333

Contents

gdxrrw-package	2
gams	2
gdxInfo	3
gdxrrw	4
igdx	4
rgdx	5
transport	8
wgdx	9
wgdx.reshape	10

Index	12
--------------	-----------

gdrrw-package

Exchanging data between GAMS and R

Description

This package implements an interface between GAMS and R. It includes functions to transfer data between GDX (the GAMS data format) and R, a function to call GAMS from within R, query functions for the meta-data in GDX, and other related utilities.

Details

This package contains five classes of functions:

- `igdx(...)` gives information on the linkage between this package and the GDX library.
- `rgdx("gdxfile", ...)` and related functions read from GDX
- `wgdx("gdxfile", ...)` and related functions write to GDX
- `gams("gmsfile and args", ...)` runs gams with the arguments provided
- `gdxInfo("gdxfile", ...)` dumps GDX content or returns GDX metadata (list of symbols, etc.)

Author(s)

Original coding by Rishabh Jain. Adopted, packaged, and extended by Steve Dirkse. Maintainer: <R@gams.com>

gams

Run a GAMS model from R

Description

Run a GAMS model from R.

Usage

```
gams(gmsAndArgs)
```

Arguments

`gmsAndArgs` Name of .gms file to run, with possible args

Value

Return from executing `gams gmsAndArgs`

Note

A common problem is failure to find the GAMS system directory. Use [igdx](#) to troubleshoot and solve this problem.

Author(s)

Original coding by Rishabh Jain. Adopted and packaged by Steve Dirkse. Maintainer: <R@gams.com>

See Also

[igdx](#), [rgdx](#), [wgdx](#)

Examples

```
## Not run:
  gams("transport.gms")
  gams("myModel.gms lp=SOPLEX --JOB_ID=case00")

## End(Not run)
```

gdxInfo

Display or Return Information from a GDX File

Description

Display all the information contained in a GDX file, along with some version & debugging info. The display format is copied from gdxdump.

Alternately, return information about the symbols contained in a GDX file in list or a more extensive data frame format.

Usage

```
gdxInfo(gdxName = NULL, dump=TRUE, returnList=FALSE, returnDF=FALSE)
```

Arguments

gdxName	the name of the GDX file. If this argument is omitted, we display only the version info for the GDX library used
dump	if TRUE, dump GDX contents to the console
returnList	if TRUE, return brief information about the symbols in the GDX file as a list of lists
returnDF	if TRUE, return extended information about the symbols in the GDX file as a list of data frames

Note

A common problem is failure to load the external GDX libraries that are required to interface with GDX data. Use [igdx](#) to troubleshoot and solve this problem.

Author(s)

Original coding by Rishabh Jain. Adopted and packaged by Steve Dirkse. Maintainer: <R@gams.com>

See Also

[igdx](#), [rgdx](#), [wgdx](#)

Examples

```
gdxInfo();
## Not run:
  gdxInfo("transport.gdx");
  gdxInfo("transport.gdx", dump=FALSE, returnDF=TRUE);

## End(Not run)
```

gdxrrw	<i>Constants used in gdxrrw</i>
--------	---------------------------------

Description

Constants used in gdxrrw inputs/outputs, e.g. to indicate variable or equation type.

igdx	<i>Initialize the External GDX Libraries</i>
------	--

Description

Initialize (i.e. load) the external GDX libraries that are required to interface with GDX data, and report on the status of the initialization.

Usage

```
igdx(gamsSysDir=NULL, silent=FALSE, returnStr=FALSE)
```

Arguments

gamsSysDir	a directory containing the external GDX libraries, typically the GAMS system directory. If this argument is omitted, no loading is done but we still report on the initialization status.
silent	controls logging of results
returnStr	controls what to return. If FALSE, return TRUE if the external GDX libraries are loaded, FALSE o/w. If TRUE, return the path to the external GDX libraries if loaded, an empty string otherwise.

Details

To query but not modify the current GDX library binding, leave the `gamsSysDir` argument `NULL`. To clear, reload, and then query the GDX library binding, pass a string argument to `gamsSysDir`.

If `gamsSysDir` is a non-empty string containing a valid directory, it will be tried first. If the GDX libraries cannot be loaded from this location, the next load attempt will make use of the environment variable `R_GAMS_SYSDIR`, provided it is set to a non-empty string. If we are still not successful, as a last resort we try to load the GDX libraries using the system-specific library search mechanism (e.g. the `PATH` on Windows or `LD_LIBRARY_PATH` on Linux).

Value

By default (when `returnStr` is `FALSE`), the return value is `TRUE` if the external GDX libraries are loaded, `FALSE` o/w. If `returnStr` is `TRUE`, the return value is the path to the directory containing the external GDX libraries if these libraries were successfully loaded, an empty string otherwise.

Note

The directory containing the external GDX libraries is also where we look for the `gams` executable, so calling `igdx` prior to calling `gams` ensures that the `gams` executable can be located. Consider saving a `.First` function like the following to your R workspace: `.First <- function() { library(gdxrrw) ; igdx("/your/GAMS/sysdir") }` or set the environment variable `R_GAMS_SYSDIR`.

Author(s)

Steve Dirkse. Maintainer: <R@gams.com>

See Also

[rgdx](#), [wgdx](#), [gams](#), [gdxInfo](#)

Examples

```
## Not run:
  igdx("C:\Program Files\gams23.6");
  igdx("/usr/gams/23.6.3");

## End(Not run)
  igdx();
```

rgdx

Read data from GDX into R

Description

Read one data item (also called a symbol) from GDX into R, returning it as a list. Note that GDX files contain multiple symbols (e.g. sets, parameters). Each symbol is read with a separate call.

Usage

```

# generic form - return a symbol as a list
rgdx(gdxName, requestList = NULL, squeeze = TRUE, useDomInfo = TRUE,
     followAlias = TRUE)

# return a set or parameter in a data frame
rgdx.set(gdxName, symName, names=NULL, compress=FALSE, ts=FALSE,
         useDomInfo = TRUE, check.names = TRUE, te = FALSE)
rgdx.param(gdxName, symName, names=NULL, compress=FALSE, ts=FALSE,
           squeeze=TRUE, useDomInfo = TRUE, check.names = TRUE)

# return a scalar
rgdx.scalar(gdxName, symName, ts=FALSE)

```

Arguments

gdxName	the name of the GDX file to read
requestList	the name of the symbol to read, and (optionally) information about how much information to return and in what format. This argument must be a named list. If omitted, the universe of UELs contained in the GDX file is returned
squeeze	if TRUE/nonzero, squeeze out any zero or EPS stored in the GDX container
useDomInfo	if TRUE, the default filter will be the domain info in the GDX. If no domain info is available, or if useDomInfo=FALSE, the default filter will be the GDX universe
followAlias	if TRUE and the symbol queried is an alias, return information for the real set rather than the alias
check.names	If TRUE then the names of the variables in the data frame are checked to ensure that they are syntactically valid variable names and are not duplicated. If necessary they are adjusted (by make.names so that they are
symName	the name of the GDX symbol to read
names	the column names to use in the data frame returned
compress	if TRUE, compress the factors in the data frame so they only include required levels. For the default compress=FALSE, each factor includes levels for the entire universe of UELs in the GDX file
ts	if TRUE, include the .ts field (i.e. the explanatory text) for the symbol in the return
te	if TRUE, include the associated text (i.e. the .te field) for each set element in the return

Details

The requestList argument to rgdx is essentially a list of arguments specifying what symbol to read and how to read it. Valid list elements are:

name name of symbol to read from GDX

- form** specify representation to use on return: “sparse” (default) or “full”
- uels** UEL filter to use when reading
- field** specify field to read for equations and variables
- te** if true, return the associated text (i.e. the `.te` field) for each set element in the return
- ts** if true, include the explanatory text (i.e. the `.ts` field) for the symbol in the return
- compress** if true, compress UEL lists of return value by removing unused elements in each index position
- dim** expected dimension of symbol to be read

When reading sets, one can specify that the associated text is included in the return value. The value returned for set elements where no associated text exists is controlled via the `options()` mechanism. Setting `options(gdx.inventSetText=NA)` (the default) returns NA, setting `options(gdx.inventSetText=T)` returns a string made up from the UEL(s), and setting `options(gdx.inventSetText=F)` returns an empty string `""`.

When reading GDX data into data frames (e.g. with `rgdx.param`), the `names()` (i.e. the column names) of the output data frame can be passed in via the optional `names` argument. If not, then the names are either created internally or taken from the domain information and name of the symbol in question. The latter choice is controlled by setting `options(gdx.domainNames=FALSE)` to use internally generated names for the data frame columns (e.g. `"i", "j"` or `"i1", "i2", "i3", "i4"`). A setting of `TRUE` (the default) means use the domain names from the GDX file for the column names.

Value

By default, the return value is a list with elements describing the data item or symbol returned. Elements include:

- name** symbol name
- type** symbol's data type: set, parameter, variable or equation
- dim** symbol dimension
- val** array containing the symbol data
- form** form of the data in `val`, i.e. full or sparse
- uels** vector of UEL lists, one list per symbol dimension
- domains** character vector of length `dim` containing the symbol's domain info
- te** (optional) associated text for sets

The functions `rgdx.param` and `rgdx.set` are special-purpose wrappers that read parameters and sets, respectively, and return them as data frames.

The function `rgdx.scalar` returns a scalar (i.e. a 0-dimensional parameter) as a double.

Note

A common problem is failure to load the external GDX libraries that are required to interface with GDX data. Use [igdx](#) to troubleshoot and solve this problem.

Author(s)

Original coding by Rishabh Jain. Adopted and packaged by Steve Dirkse. Maintainer: <R@gams.com>

See Also

[igdx](#), [wgdx](#), [gdxInfo](#)

Examples

```
# run R-script transport.r from the data subdirectory of the gdxrrw package
# to load up some data for writing to a GDX file
data(transport)

wgdx("rgdx1",sf,si,sj,sa,sb,sd)
ou = rgdx("rgdx1")
req <- list(name="f")
of = rgdx("rgdx1",req)
req <- list(name="i")
oi = rgdx("rgdx1",req)
req <- list(name="j")
oj = rgdx("rgdx1",req)
req <- list(name="a")
oa = rgdx("rgdx1",req)
req <- list(name="b")
ob = rgdx("rgdx1",req)
req <- list(name="d")
od = rgdx("rgdx1",req)

## Not run:
# complete tests and examples can be run in the
# extdata directory of the gdxrrw package
# check .libPaths for a hint on where packages are installed
setwd(paste(.libPaths()[1],"/gdxrrw/extdata",sep=""))
source("tAll.R")

## End(Not run)
```

transport

Sample datasets

Description

Datasets used to illustrate wgdx function.

Usage

```
data(transport)
```


Description

Create a GDX file containing the GAMS data described in the lists Note that each list describes a separate data element - i.e. a set or parameter - all of which are written to a single file.

Usage

```
# generic form - each arg is a list specifying a symbol to write
wgdx(gdxName, ..., squeeze = 'y')

# write multiple symbols specified in list, data frame, or scalar form
wgdx.lst(gdxName, ..., squeeze = 'y')
```

Arguments

gdxName	the name of the GDX file to write
...	zero or more arguments describing the symbols to write. For <code>wgdx</code> , each arg must be a list holding a symbol (e.g. as returned by <code>rgdx</code>). For <code>wgdx.lst</code> , each arg is also allowed to be a data frame holding a symbol (e.g. as returned by <code>rgdx.param</code> or <code>rgdx.set</code>), a scalar holding a symbol (e.g. as returned by <code>rgdx.scalar</code>), or a list containing any combination of these elements.
squeeze	if 'y'/TRUE/nonzero, squeeze out zeros: do not store in GDX. If 'n'/FALSE/zero, do not squeeze out zeros: store explicit zeros in GDX. If 'e', store zeros as EPS in the GDX.

Note

A common problem is failure to load the external GDX libraries that are required to interface with GDX data. Use [igdx](#) to troubleshoot and solve this problem.

Author(s)

Original coding by Rishabh Jain. Adopted and packaged by Steve Dirkse. Maintainer: <R@gams.com>

See Also

[igdx](#), [rgdx](#), [gdxInfo](#)

Examples

```
data(trnsport);
wgdx("wgdx1.gdx", sf, si, sj, sa, sb, sd);

## Not run:
# complete tests and examples can be run in the
```

```
# extdata directory of the gdxrrw package
# check .libPaths for a hint on where packages are installed
setwd(paste(.libPaths()[1], "/gdxrrw/extdata", sep=""))
source("tAll.R")

## End(Not run)
```

wgdx.reshape

Reshape R data to prepare it for writing to GDX

Description

Reshape the input data frame (assumed to be in "wide" form) into a "long" form suitable for passing to `wgdx.lst` or `wgdx.df`.

Usage

```
wgdx.reshape(inDF, symDim, symName=NULL, tName="time",
            .gdxName=NULL, setsToo=TRUE, order=NULL,
            .setNames=NULL)
```

Arguments

<code>inDF</code>	data frame in "wide" form
<code>symDim</code>	dimension of the output GDX symbol
<code>symName</code>	name of the output GDX symbol
<code>tName</code>	index set name for the new index position created by reshaping
<code>.gdxName</code>	name of the GDX file to write
<code>setsToo</code>	if TRUE, extract the index sets defined by <code>inDF</code>
<code>order</code>	specify the selection and ordering of the index columns when reshaping
<code>setNames</code>	specify explanatory text for the extracted index sets

Note

A common problem is failure to load the external GDX libraries that are required to interface with GDX data. Use [igdx](#) to troubleshoot and solve this problem.

There is also the issue of our dependence on the reshape package. This still needs to be tightened up.

This routine is experimental/prototype work and perhaps temporary. Don't use it for production work.

Author(s)

Original coding by Steve Dirkse. Maintainer: <R@gams.com>

See Also[wgdx](#), [igdx](#), [rgdx](#), [gdxInfo](#)**Examples**

```
# take a sample dataset and reshape it
str(airquality);
oList <- wgdx.reshape (airquality, 3, symName="airquality",
                      tName = "dataType", order=c(5,6,0))

## Not run:
# send the data to GDX
wgdx.lst("airquality", oList)

## End(Not run)

## Not run:
# complete tests and examples can be run in the
# extdata directory of the gdxrrw package
# check .libPaths for a hint on where packages are installed
setwd(paste(.libPaths()[1],"/gdxrrw/extdata",sep=""))
source("tAll.R")

## End(Not run)
```

Index

- * **datasets**
 - gdxrrw, 4
 - transport, 8
 - * **data**
 - gams, 2
 - gdxInfo, 3
 - gdxrrw-package, 2
 - igdx, 4
 - rgdx, 5
 - wgdx, 9
 - wgdx.reshape, 10
 - * **interface**
 - gams, 2
 - gdxInfo, 3
 - gdxrrw-package, 2
 - igdx, 4
 - rgdx, 5
 - wgdx, 9
 - wgdx.reshape, 10
 - * **optimize**
 - gams, 2
 - gdxInfo, 3
 - gdxrrw-package, 2
 - igdx, 4
 - rgdx, 5
 - wgdx, 9
 - wgdx.reshape, 10
 - * **package**
 - gdxrrw-package, 2
- d (transport), 8
- gams, 2, 5
gdxInfo, 3, 5, 8, 9, 11
gdxrrw, 4
gdxrrw (gdxrrw-package), 2
gdxrrw-package, 2
GMS_EQUTYPE (gdxrrw), 4
GMS_VARTYPE (gdxrrw), 4
igdx, 3, 4, 4, 7-11
rgdx, 3-5, 5, 9, 11
sa (transport), 8
sb (transport), 8
sd (transport), 8
sf (transport), 8
si (transport), 8
sj (transport), 8
transport, 8
uels (transport), 8
wgdx, 3-5, 8, 9, 11
wgdx.reshape, 10