

Package ‘TSSQLite’

October 27, 2009

Version 2009.10-1

Date 2009-10-14

Title Time Series Database Interface extentions for SQLite

Description TSSQLite provides a SQLite interface for TSdbi.

Depends R (>= 2.8.0), methods, tframePlus, TSdbi, RSQLite

Imports methods, DBI, TSdbi (>= 2009.5-1), RSQLite

Suggests zoo, tseries, TShistQuote

Enhances TSdbi

License GPL-2

Author Paul Gilbert <pgilbert@bank-banque-canada.ca>

Maintainer Paul Gilbert <pgilbert@bank-banque-canada.ca>

URL <http://www.bank-banque-canada.ca/pgilbert>

Repository CRAN

Repository/R-Forge/Project tsdbi

Repository/R-Forge/Revision 91

Date/Publication 2009-10-27 07:23:18

R topics documented:

TSdbiMethods	2
Index	5

TSdbiMethods

TSdbi SQLite Methods

Description

Methods for TSdbi SQLite time series database connection.

Usage

```
## S4 method for signature 'SQLiteDriver, character':
TSconnect(drv, dbname, ...)
## S4 method for signature 'character,
## SQLiteConnection':
TSget(serIDs, con,
      TSrepresentation=options()$TSrepresentation,
      tf=NULL, start=tfstart(tf), end=tfend(tf),
      names=serIDs, TSdescription=FALSE, TSdoc=FALSE, TSlabel=FALSE,
      vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'ANY, character,
## SQLiteConnection':
TSput(x, serIDs, con, Table=NULL,
      TSdescription.=TSdescription(x), TSdoc.=TSdoc(x), TSlabel.=TSlabel(x),
      vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,
## SQLiteConnection':
TSdates(serIDs, con,
        vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,
## SQLiteConnection':
TSdelete(serIDs, con,
         vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,
## SQLiteConnection':
TSdescription(x, con, ...)
## S4 method for signature 'character,
## SQLiteConnection':
TSdoc(x, con, ...)
## S4 method for signature 'character,
## SQLiteConnection':
TSlabel(x, con, ...)
## S4 method for signature 'TSSQLiteConnection':
print(x, ...)
```

Arguments

drv	A SQLiteDriver.
dbname	A character string indicating a database name.

<code>con</code>	A database connection.
<code>serIDs</code>	identifiers for series to extract.
<code>x</code>	data series to put on the database, or a series identifier for <code>TSdescription</code> and <code>TSdoc</code> or, for print, a database connection as returned by <code>TSconnect</code> .
<code>TSrepresentation</code>	time series representation to use for the result.
<code>names</code>	Optional character vector to use for series names.
<code>tf</code>	Optional tframe to use for truncating data. (See <code>tfwindow</code> .)
<code>start</code>	Optional start date to use for truncating data. (See <code>tfwindow</code> .)
<code>end</code>	Optional end date to use for truncating data. (See <code>tfwindow</code> .)
<code>TSdescription</code>	TRUE or FALSE indicating whether description should be extracted
<code>TSdescription.</code>	Description to put on the database.
<code>TSdoc</code>	TRUE or FALSE indicating whether documentation should be extracted.
<code>TSdoc.</code>	Documentation to put on the database.
<code>TSlabel</code>	TRUE or FALSE indicating whether series label should be extracted.
<code>TSlabel.</code>	Short series label to put on the database.
<code>Table</code>	Database table indication (necessary if it cannot be determined automatically).
<code>vintage</code>	Vintage to be supplied (if supported by db).
<code>panel</code>	Panel to be supplied (if supported by db).
<code>...</code>	Arguments passed to other methods.

Details

These are SQLite methods for **TSdbi**. See the **TSdbi** for details and see the vignette distributed with this package for more complete examples.

Value

depends.

See Also

[TSdbi-package](#), [dbConnect](#), [TSput](#), [TSget](#)

Examples

```
con <- try(TSconnect(dbDriver("SQLite"), dbname="test"))
if(! inherits(con, "try-error")) {
  z <- ts(rnorm(100), start=c(1975,1), frequency=12)
  seriesNames(z) <- "random1"
  if(TSexists("random1", con)) TSreplace(z, con) else
  TSput(z, con)
  z1 <- TSget("random1", con)
```

```

tfplot(z1)
z <- ts(matrix(rnorm(100),50,2), start=c(1975,1), frequency=4)
seriesNames(z) <- c("random2","random3")
if(TSexists("random2", con) |
    TSexists("random3", con) ) TSreplace(z, con) else
TSput(z, con)
z2 <- TSget("random2", con)
tfplot(z2)
TSdates("D1", con)
TSdates("random2", con)
TSdates(c("random1","random2","random3"), con)
TSmeta("random2", con)
options(TSconnection=con)
z2 <- TSget(c("random2","random3"))
z <- TSdates(c("D1","random2","random3"))
print(z)
TSmeta("random2")
TSdelete("random1", con)
TSdelete("random2")
}
con <- try(TSconnect(dbDriver("SQLite"), dbname="ets"))
if(! inherits(con, "try-error")) {
  TSmeta("B103", con)
  z1 <- TSget("B103", con)
  tfplot(z1)
  z2 <- TSget(c("B103", "B104"), con)
  tfplot(z2)
  options(TSconnection=con)
  TSmeta("B103")
  z2 <- TSget(c("B103","B104"))
  TSmeta(z2)
  TSdates("D1", con)
  TSdates("B103", con)
  TSdates(c("D1","B103","B104"), con)
  z <- TSdates(c("D1","B103","B104"))
  print(z)
  start(z)
  end(z)
}

```

Index

*Topic **ts**

- TSdbiMethods, [2](#)
- dbConnect, [3](#)
- print, TSSQLiteConnection-method
(*TSdbiMethods*), [2](#)
- TSconnect, SQLiteDriver, character-method
(*TSdbiMethods*), [2](#)
- TSdates, character, SQLiteConnection-method
(*TSdbiMethods*), [2](#)
- TSdbi-package, [3](#)
- TSdbiMethods, [2](#)
- TSdelete, character, SQLiteConnection-method
(*TSdbiMethods*), [2](#)
- TSdescription, character, SQLiteConnection-method
(*TSdbiMethods*), [2](#)
- TSdoc, character, SQLiteConnection-method
(*TSdbiMethods*), [2](#)
- TSget, [3](#)
- TSget, character, SQLiteConnection-method
(*TSdbiMethods*), [2](#)
- TSlabel, character, SQLiteConnection-method
(*TSdbiMethods*), [2](#)
- TSput, [3](#)
- TSput, ANY, character, SQLiteConnection-method
(*TSdbiMethods*), [2](#)
- TSSQLiteConnection-class
(*TSdbiMethods*), [2](#)