

Package ‘TSdbi’

June 10, 2009

Version 2009.5-2

Date 2009-5-27

Title Time Series Database Interface

Description TSdbi provides a common interface to time series databases. The objective is to define a standard interface so users can retrieve time series data from various sources with a simple, common, set of commands, and so programs can be written to be portable with respect to the data source. The SQL implementations also provide a database table design, so users needing to set up a time series database have a reasonably complete way to do this easily. The interface provides for a variety of options with respect to the representation of time series in R. There is also a (not yet well tested) mechanism to handle multilingual data documentation.

Depends R (>= 2.8.0), methods, tframe (>= 2008.5-1)

Imports methods, DBI

Suggests zoo, tseries

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 72

Date/Publication 2009-06-10 08:04:48

R topics documented:

TSdbi-package	2
TScheckdbi	3
TSconnect	4
TSdates	5
TSdescription	6
TSfinddb	8
TSget	9
TSput	10
TSquery	11
TSsourceInfo	12

Index	14
--------------	-----------

TSdbi-package	<i>Time Series Data Base Interface</i>
---------------	--

Description

TSdbi provides an common interface to time series databases. The package can use the DBI package and its interface to SQL databases, in which case a table structure is specified by TSdbi. It can also be used as an interface to Fame databases (through the padi protocol for now, but directly should be possible with some work).

Details

Package: TSdbi
 Depends: R (>= 2.5.0), GPArotation, setRNG (>= 2004.4-1), tframe (>= 2007.5-2), dse1 (>= 2007.5-2), dse2 (>= 2007.5-2)
 Suggests: CDNmoney
 License: GPL Version 2.
 URL: <http://www.bank-banque-canada.ca/pgilbert>

The main functions are:

TSconnect	Connect to a database.
TSget	Extract a series from a database.
TSput	Write a series to the database.
TSdates	Check the availability of a series.
TSdescription	Extract the long description of a series.
TSdoc	Extract the documentation for a series.

Use of this package requires one of the interface packages (e. g. **TSSQLite**, **TSMYSQL**, **TSpadi**) An overview of how to use the package is (will be) available in vignettes. Using **TSdbi** is very similar for the different database interfaces, but building vignettes requires working code so the vignettes are included in the various interface packages. For the same reason, most examples and

demos must be included in the interface packages. Consult the documentation for the methods in a particular interface package for most examples.

Options can be set to simplify access to a commonly used database (see [TSput](#)).

Author(s)

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

See Also

[TSconnect](#), [TSget](#), [TSput](#), [TSdates](#), [dbConnect](#), [TSdbiMethods](#), [TSdbiMethods](#), [TSdbiMethods](#),

TScheckdbi

Check Connection

Description

Check if time series database connection is ok.

Usage

```
TScheckdbi(con)
## S4 method for signature 'missing':
TScheckdbi(con=getOption("TSconnection"))
## S4 method for signature 'ANY':
TScheckdbi(con=getOption("TSconnection"))
```

Arguments

`con` A database connection.

Details

The function `TScheckdbi` checks if a connection to a server is a connection to a valid time series database.

Value

TRUE or FALSE.

See Also

[dbConnect](#), [TSdates](#), [TSget](#), [TSput](#)

TSconnect

*Connect to a Time Series Database***Description**

Return a connection to a time series database

Usage

```
TSconnect(drv, dbname, ...)
## S4 method for signature 'character, character':
TSconnect(drv, dbname, ...)
## S4 method for signature 'logicalId':
show(object)
## S4 method for signature 'TSdb':
show(object)
## S4 method for signature 'TSdb':
print(x, ...)
```

Arguments

drv	A database connection driver or character string.
dbname	The name of the database to which the connection should be established
x	A database connection as returned by TSconnect.
object	an object to display.
...	Additional arguments passed to other print methods.

Details

This function establishes a connection using a driver from one of the driver packages (e.g. **TSMysql** or **TSSQLite**). If `drv` is a character string (e.g. "MySQL") then the method attempts to get a driver using the character string.

"TSconnect" uses `dbConnect` but checks the database has expected tables and also establishes information about additional features that may be available (vintiges and/or panels).

Options can be set to simplify access to a commonly used database (see [TSput](#)).

Value

A database connection.

See Also

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

TSdates

Check Data Availability

Description

Check the dates for which date is available.

Usage

```
TSdates (serIDs, con=getOption("TSconnection"), ...)  
  
## S4 method for signature 'character, missing':  
TSdates (serIDs, con=getOption("TSconnection"), ...)  
## S4 method for signature 'character, ANY':  
TSdates (serIDs, con=getOption("TSconnection"), ...)  
  
## S3 method for class 'TSdates':  
start(x, ...)  
## S3 method for class 'TSdates':  
tfstart(x)  
## S3 method for class 'TSdates':  
end(x, ...)  
## S3 method for class 'TSdates':  
tfend(x)
```

Arguments

<code>con</code>	A database connection.
<code>serIDs</code>	identifiers for series on the database.
<code>x</code>	an object returned by <code>TSdates</code> .
<code>...</code>	arguments passed to other methods.

Details

`TSdates` returns information about the start and end of each series in `serIDs`. `con` is a database connection as returned by `dbConnect`. `TSdates` also provides simple way to query a regularly used database. The connection can be set in options using `options(TSconnection=con)` and then only the series identifiers need to be specified in calls to `TSdates`.

`start`, `tfstart`, `end`, and `tfend` extract start and end dates from the object returned by `TSdates`.

Value

depends.

See Also

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

TSdescription

Specific Methods for Documenting Data

Description

See the generic function description.

Usage

```

TSexists(serIDs, con=getOption("TSconnection"), ...)
## S4 method for signature 'default':
TSexists(serIDs, con=getOption("TSconnection"), ...)

TSMeta(x, con, ...)
## S4 method for signature 'character, missing':
TSMeta(x, con=getOption("TSconnection"), ...)
## S4 method for signature 'character, ANY':
TSMeta(x, con=getOption("TSconnection"), ...)
## S4 method for signature 'ANY, missing':
TSMeta(x, con, ...)

TSMeta(x) <- value

TSdescription(x, con, ...)
## S4 method for signature 'character, missing':
TSdescription(x, con=getOption("TSconnection"), ...)
## S4 method for signature 'character, ANY':
TSdescription(x, con=getOption("TSconnection"), ...)
## S4 method for signature 'ANY, missing':
TSdescription(x, con, ...)
## S4 method for signature 'missing, ANY':
TSdescription(x, con, serIDs, ...)
## S4 method for signature 'missing, missing':
TSdescription(x, serIDs, ...)
TSdescription(x) <- value

TSDoc(x, con, ...)
## S4 method for signature 'character, missing':
TSDoc(x, con=getOption("TSconnection"), ...)
## S4 method for signature 'character, ANY':
TSDoc(x, con=getOption("TSconnection"), ...)
## S4 method for signature 'ANY, missing':
TSDoc(x, con, ...)

```

```

## S4 method for signature 'missing, ANY':
TSdoc(x, con, serIDs, ...)
## S4 method for signature 'missing, missing':
TSdoc(x, serIDs, ...)
TSdoc(x) <- value

TSlabell(x, con, ...)
## S4 method for signature 'character, missing':
TSlabell(x, con=getOption("TSconnection"), ...)
## S4 method for signature 'character, ANY':
TSlabell(x, con=getOption("TSconnection"), ...)
## S4 method for signature 'ANY, missing':
TSlabell(x, con, ...)
## S4 method for signature 'missing, ANY':
TSlabell(x, con, serIDs, ...)
## S4 method for signature 'missing, missing':
TSlabell(x, serIDs, ...)
TSlabell(x) <- value

TSrefperiod(x)
## S4 method for signature 'default':
TSrefperiod(x)
TSrefperiod(x) <- value

```

Arguments

<code>con</code>	A database connection.
<code>serIDs</code>	identifiers for series on the database.
<code>x</code>	a time series data object or an identifier for a series on the database.
<code>value</code>	a character string (or vector of character strings).
<code>...</code>	arguments passed to other methods.

Details

These functions return various information about the data series. Methods with `con` and `serIDs` (sometimes identifier are passed as argument `x`) get data from the database. Others extract information from the object.

`TSexists` returns `TRUE` or `FALSE`, depending on whether the series exist at the connection. (All series specified must exist for `TRUE`.)

Assignments assign an attribute to the object `x` with `value`. If `x` is a multivariate time series (matrix) then `value` should be a vector of length equal the number of series. The reference period for a time series indicates a special refence point (e.g. "Wednesday" for weekly data collected on Wednesday).

The extraction methods extract the attribute.

Value

Depends. See details.

See Also

[TSget](#) [TSput](#) [TSdates](#)

TSfinddb

Find a Time Series Database Connection

Description

Find a connection to a specified time series database.

Usage

```
TSfinddb(dbname=NULL, driverOrder=c("MySQL", "SQLite", "padi"))
```

Arguments

dbname	Character string indicating the name of a database.
driverOrder	A vector of character string indicating TSdbi drivers in the order they should be tried.

Details

TSfinddb tries to establish a connection to the indicated database using the drivers in the order specified. This attempt also requires the corresponding TSdbi driver package (e.g., "TSMysql", "TSSQLite", or "TSpadi"). If the package cannot be loaded then the driver is skipped. The first valid connection is returned. If no valid connection is found then NULL is returned.

Value

A connection

See Also

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

Description

Get time series matrix structure from a database

Usage

```
TSget (serIDs, con=getOption("TSconnection"), ...)
## S4 method for signature 'character, missing':
TSget (serIDs, con=getOption("TSconnection"), ...)
## S4 method for signature 'character, ANY':
TSget (serIDs, con=getOption("TSconnection"), ...)
```

Arguments

<code>con</code>	A database connection.
<code>serIDs</code>	identifiers for series to extract.
<code>...</code>	Arguments passed to <code>TSgetSQL</code> or other methods.

Details

These functions extract data from a database using a connection. This method is generic. The argument `serIDs` should give identifiers for the series to extract.

`TSget` and other functions also provide a way to query a regularly used database by setting the connection in options using `options(TSconnection=con)`, so then only the series identifiers need to be specified in calls to `TSget`.

The user can specify a default time series representation with the argument `TSrepresentation="something"` where "something" is "default" by default, but might be "zoo" or something else which is used to coerce the representation. The `TSrepresentation` is passed in the `...` argument. The representation is applied using `do.call(TSrepresentation, list(mat))` where `mat` is the time series (matrix) to be return. If `TSrepresentation` is not specified then the `ts` representation is used for data from tables "A", "Q", "M", "S" and `zoo` otherwise. See [TSput](#) for a list of the various tables.

It would be possible to specify `TSrepresentation="as.zoo"`, but this may result in `as.zoo` being applied twice, in which case some information about the time representation gets lost, so the best way to get a `zoo` representation is to specify `TSrepresentation="zoo"`.

Users can set a session default with `options(TSrepresentation="something")` so that this is always passed as an argument to `TSget`.

It is also possible to pass `start`, `end`, or `tframe` information to truncate the returned series. This is part of the `...` argument passed to `tfwindow`. See [tfwindow](#) for more details. By default no truncation is applied.

If the database supports vintages or panels then it is also possible to set defaults for these with, for example, `options(TSvintage="current")` and `options(TSpanel="Canada")`. The default specification has to be supported by the database for this to work.

`names`, `TSdescription`, `TSdoc` and `TSlabel` can also be specified as arguments. (Passed in ...). See [TSputSQL](#) for more details.

Value

A time series matrix.

See Also

[TSdbi-package](#), [TSconnect](#), [TSput](#), [TSdates](#) [tfwindow](#)

TSput

Write Data to a Data Connection

Description

Write data to a server.

Usage

```
TSput(x, serIDs=seriesNames(x), con=getOption("TSconnection"), ...)
TSdelete(serIDs, con=getOption("TSconnection"), ...)
TSreplace(x, serIDs=seriesNames(x), con=getOption("TSconnection"), ...)

## S4 method for signature 'ANY, missing, missing':
TSput(x, serIDs=seriesNames(x), con=getOption("TSconnection"), ...)
## S4 method for signature 'ANY, DBIConnection,
## missing':
TSput(x, serIDs=seriesNames(x), con=getOption("TSconnection"), ...)
## S4 method for signature 'ANY, character, ANY':
TSput(x, serIDs=seriesNames(x), con=getOption("TSconnection"), ...)
## S4 method for signature 'character, missing':
TSdelete(serIDs, con=getOption("TSconnection"), ...)
## S4 method for signature 'character, ANY':
TSdelete(serIDs, con=getOption("TSconnection"), ...)
## S4 method for signature 'default':
TSreplace(x, serIDs=seriesNames(x), con=getOption("TSconnection"), ...)
```

Arguments

<code>con</code>	A database connection.
<code>x</code>	time series data.
<code>serIDs</code>	identifiers for series on the database.
<code>...</code>	Arguments passed to other methods.

Details

Class `logicalId` is a logical indicating if the operation succeeded, and also contains meta data indicating how to retrieve the data. (Except in the case of `TSdelete` the data cannot be retrieved.)

These functions write data to a database connection. `TSreplace` removes any existing object first. `TSput` will fail if a series with the same identifier already exists.

`TSput` and `TSreplace` provide ways to query a regularly used single database. The connection can be set in options using `options(TSconnection=con)` and then only the series identifiers need to be specified in calls to `TSput` and `TSreplace`.

`TSdescription` and `TSdoccan` also be set. (Passed in ...). See [TSputSQL](#) for more details.

If an appropriate table cannot be determined from the series it will be necessary to pass the `Table` argument (in ...). The DBI/SQL interface uses the following tables:

A	for annual data
Q	for quarterly data
M	for monthly data
S	for semiannual data
W	for weekly data
D	for daily data
B	for business data
U	for minutely data
I	for irregular data with a date
T	for irregular data with a date and time
Meta	for meta data

Value

An object of class `logicalId`.

See Also

[TSdbi-package](#), [TSdates](#), [TSget](#), [TSputSQL](#), [dbConnect](#)

TSquery

Construct Time Series Data From a Database

Description

Construct a time series from a database

Usage

```
TSquery(select, dateField, table, where=NULL, frequency="monthly",
        na.as=0, names=NULL, con=options()$connection)
```

Arguments

<code>select</code>	character string used to construct query.
<code>dateField</code>	character string used to construct query.
<code>table</code>	character string used to construct query.
<code>where</code>	character string used to construct query.
<code>frequency</code>	character string used to specify frequency of the result (only "daily", "monthly" or "annual" are currently supported).
<code>na.as</code>	value to be used to replace NAs in the returned series.
<code>names</code>	optional vector of character strings to be used for the returned series. If not NULL it must have length equal to the number of series returned.
<code>con</code>	A database connection.

Details

This functions is unlike other functins in this package (and may eventually be moved to a separate package). It constructs a time series from a database (using a connection) and a query generated with the function arguments. An example query might be something like `z <- TSquery(select="SUM(amount)", dateField="issue_date", table="term", where="term > 10", frequency="annual", con=con)`

Value

A time series or time series matrix.

See Also

[TSget](#),

TSsourceInfo	<i>Get source information from a data object</i>
--------------	--

Description

Get source information from an object

Usage

```
TSseriesIDs(x)
TScon(x)
TSextractionDate(x)
```

Arguments

`x` An object which contains source series information (as returned by `TSget`).

Value

Strings indicating the information.

See Also

[TGet](#), [TConnect](#), [TSdates](#)

Index

*Topic **package**

TSdbi-package, 2

*Topic **ts**

TScheckdbi, 3

TSconnect, 3

TSdates, 4

TSdbi-package, 2

TSdescription, 6

TSfinddb, 8

TSget, 8

TSput, 10

TSquery, 11

TSsourceInfo, 12

conType-class (*TSconnect*), 3

dbConnect, 2–5, 8, 11

end.TSdates (*TSdates*), 4

logicalId-class (*TSput*), 10

print, TSdb-method (*TSconnect*), 3

show, logicalId-method
(*TSconnect*), 3

show, TSdb-method (*TSconnect*), 3

start.TSdates (*TSdates*), 4

tfend, 5

tfend.TSdates (*TSdates*), 4

tfstart, 5

tfstart.TSdates (*TSdates*), 4

tfwindow, 9

TScheckdbi, 3

TScheckdbi, ANY-method
(*TScheckdbi*), 3

TScheckdbi, missing-method
(*TScheckdbi*), 3

TScon (*TSsourceInfo*), 12

TSconnect, 2, 3, 9, 12

TSconnect, character, character-method
(*TSconnect*), 3

TSdates, 2, 3, 4, 4, 7–9, 11, 12

TSdates, character, ANY-method
(*TSdates*), 4

TSdates, character, missing-method
(*TSdates*), 4

TSdb-class (*TSconnect*), 3

TSdbi-package, 4, 5, 8, 9, 11

TSdbi-package, 2

TSdbi.Intro (*TSdbi-package*), 2

TSdbiMethods, 2

TSdelete (*TSput*), 10

TSdelete, character, ANY-method
(*TSput*), 10

TSdelete, character, missing-method
(*TSput*), 10

TSdescription, 5, 6

TSdescription, ANY, missing-method
(*TSdescription*), 6

TSdescription, character, ANY-method
(*TSdescription*), 6

TSdescription, character, missing-method
(*TSdescription*), 6

TSdescription, missing, ANY-method
(*TSdescription*), 6

TSdescription, missing, missing-method
(*TSdescription*), 6

TSdescription<- (*TSdescription*), 6

TSdoc (*TSdescription*), 6

TSdoc, ANY, missing-method
(*TSdescription*), 6

TSdoc, character, ANY-method
(*TSdescription*), 6

TSdoc, character, missing-method
(*TSdescription*), 6

TSdoc, missing, ANY-method
(*TSdescription*), 6

TSdoc, missing, missing-method

- (TSdescription)*, 6
- TSdoc<- (*TSdescription*), 6
- TSexists (*TSdescription*), 6
- TSexists, default-method
(TSdescription), 6
- TSextractionDate (*TSsourceInfo*),
12
- TSfinddb, 8
- TSget, 2-5, 7, 8, 8, 11, 12
- TSget, character, ANY-method
(TSget), 8
- TSget, character, missing-method
(TSget), 8
- TSid-class (*TSconnect*), 3
- TSlabel (*TSdescription*), 6
- TSlabel, ANY, missing-method
(TSdescription), 6
- TSlabel, character, ANY-method
(TSdescription), 6
- TSlabel, character, missing-method
(TSdescription), 6
- TSlabel, missing, ANY-method
(TSdescription), 6
- TSlabel, missing, missing-method
(TSdescription), 6
- TSlabel<- (*TSdescription*), 6
- TSmeta (*TSdescription*), 6
- TSmeta, ANY, missing-method
(TSdescription), 6
- TSmeta, character, ANY-method
(TSdescription), 6
- TSmeta, character, missing-method
(TSdescription), 6
- TSmeta-class (*TSdescription*), 6
- TSmeta<- (*TSdescription*), 6
- TSput, 2-5, 7-9, 10
- TSput, ANY, character, ANY-method
(TSput), 10
- TSput, ANY, character, missing-method
(TSput), 10
- TSput, ANY, DBIConnection, missing-method
(TSput), 10
- TSput, ANY, missing, missing-method
(TSput), 10
- TSputSQL, 9-11
- TSquery, 11
- TSrefperiod (*TSdescription*), 6
- TSrefperiod, default-method
(TSdescription), 6
- TSrefperiod<- (*TSdescription*), 6
- TSreplace (*TSput*), 10
- TSreplace, default-method (*TSput*),
10
- TSseriesIDs (*TSsourceInfo*), 12
- TSsourceInfo, 12