

# Package ‘TSdbi’

January 2, 2012

**Version** 2011.11-2

**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, tis, tframePlus

**BuildVignettes** false

**License** GPL-2 | file LICENSE

**Author** Paul Gilbert <pgilbert.ttv9z@ncf.ca>

**Maintainer** Paul Gilbert <pgilbert.ttv9z@ncf.ca>

**URL** <http://tsdbi.r-forge.r-project.org/>

**Repository** CRAN

**Repository/R-Forge/Project** tsdbi

**Repository/R-Forge/Revision** 295

**Date/Publication** 2011-11-28 11:59:50

## R topics documented:

TSdbi-package	2
dropTStable	3
TScheckdbi	4
TSconnect	5
TSdates	6
TSdescription	7
TSfinddb	9
TSget	10
TSput	11
TSquery	13
TSsourceInfo	14
TSvintages	15
<b>Index</b>	<b>16</b>

---

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), methods, tframe (>= 2008.5-1)
Imports:    methods, DBI
Suggests:   zoo, tseries
License:    GPL Version 2.
URL:        http://tsdbi.r-forge.r-project.org

```

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.ttv9z@ncf.ca> Maintainer: Paul Gilbert <pgilbert.ttv9z@ncf.ca>

### See Also

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

---

dropTStable	<i>Drop (Remove) a Database Table</i>
-------------	---------------------------------------

---

### Description

Drop (remove) a database table if it exists.

### Usage

```
dropTStable(con, Table, yesIknowWhatIamDoing=FALSE)
```

### Arguments

con	A database connection.
Table	A character string indicating a table to drop.
yesIknowWhatIamDoing	logical to help prevent accidents.

### Details

**WARNING:** use this function only if you know what you are doing. It may destroy the whole database. The function `dropTStable` removes the indicated table if it exists on the `con`. This function will not generally be needed by users. It is used for initializing and testing the database set up, and in many cases does simply `if(dbExistsTable(con, Table)) dbRemoveTable(con, Table)`. It needs to be generic in order to work around the problem that different db engines treat capitalized table names differently. For example, MySQL uses table name `Meta` while Posgresql converts to `meta`. A default `con` is not used on purpose to help avoid accidental use. The argument `yesIknowWhatIamDoing` defaults to `FALSE` and must be set to `TRUE` or the function will return a `n` error. Beware that dropping tables will likely destroy the integrity of the of the database, and would usually only be done when you are intializing a TS database. Database permission may also be set to prevent users from dropping tables. If that is the case, then this function will return a failure error.

**Value**

TRUE

**See Also**

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

---

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](#)

---

TScnect                      *Connect to a Time Series Database*

---

## Description

Return a connection to a time series database

## Usage

```

TScnect(drv, dbname, ...)
  ## S4 method for signature 'character,character'
TScnect(drv, dbname, ...)
  ## S4 method for signature 'logicalId'
show(object)
  ## S4 method for signature 'TSdb'
show(object)
  ## S4 method for signature 'TMeta'
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 TScnect.
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.

"TScnect" 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 [TSpout](#)).

## Value

A database connection.

## See Also

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

---

 TSdates *Check Data Availability*


---

**Description**

Check the dates for which date is available.

**Usage**

```

TSdates(serIDs, con=getOption("TSconnection"),
        vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)

## S4 method for signature 'character,missing'
TSdates(serIDs, con=getOption("TSconnection"),
        vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,ANY'
TSdates(serIDs, con=getOption("TSconnection"),
        vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)

## 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**

con	A database connection.
serIDs	identifiers for series on the database.
x	an object returned by TSdates.
vintage	character string indicating vintage of the series on the database (if supported by the database).
panel	character string indicating panel of the series on the database (if supported by the database).
...	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"),
vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
  ## S4 method for signature 'default'
TSexists(
serIDs, con=getOption("TSconnection"),
vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)

  TSmeta(x, con=getOption("TSconnection"), ...)
  ## 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=getOption("TSconnection"), ...)
  ## 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=getOption("TSconnection"), ...)

```

```

    ## 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

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

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

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

```

### Arguments

con	A database connection.
serIDs	identifiers for series on the database.
vintage	character string indicating vintage of the series on the database (if supported by

	the database).
panel	character string indicating panel of the series on the database (if supported by the database).
x	a time series data object or an identifier for a series on the database.
value	a character string (or vector of character strings).
...	arguments passed to other methods.

### Details

These functions return various information about the data series. Methods with `con` and `serIDs` (sometimes identifiers 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 reference 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](#)

---

TSget

*Get Time Series Data Using a Database Connection*

---

**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**

con	A database connection.
serIDs	identifiers for series to extract.
...	Arguments passed to TSgetSQL or other methods. See details.

**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 conversion is done with the function changeTSrepresentation. If the representation is a character string then it

is applied using `do.call(TSrepresentation, list(mat, dates))` where `mat` is the time series (matrix) to be returned and `dates` are determined by `as.Date(time(x))` where `x` is the default representation of the data. If `representation` is not a character string then it should be a function and is applied using `TSrepresentation(mat, dates)`

If `TSrepresentation` is not specified, or is specified as "default", then for SQL packages (**TSMYSQL**, **TSPostgreSQL**, **TSSQLite**, etc) 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. For other packages the default is `zoo` in most cases, but may vary.

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.

Also, if the database supports vintages or panels it is possible to give a vector value for one of `vintage` or `panel` as long as `serIDs` is length 1. (That is, only one of `serIDs`, `vintage` or `panel` can have more than one element.) In this case, if `names` is not specified, `vintage` or `panel` will be used for the series names in the returned time series matrix.

`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](#) [changeTSrepresentation](#)

---

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"),
vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)

TSreplace(x, serIDs=seriesNames(x), con=getOption("TSconnection"),
vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)

## 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"),
vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,ANY'
TSdelete(
serIDs, con=getOption("TSconnection"),
vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,missing,ANY,ANY'
TSdelete(
serIDs, con=getOption("TSconnection"),
vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,ANY,ANY,ANY'
TSdelete(
serIDs, con=getOption("TSconnection"),
vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)

## S4 method for signature 'default'
TSreplace(
x, serIDs=seriesNames(x), con=getOption("TSconnection"),
vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)

```

**Arguments**

con	A database connection.
x	time series data.
serIDs	identifiers for series on the database.
vintage	character string indicating vintage of the series on the database (if supported by the database).

panel character string indicating panel of the series on the database (if supported by the database).

... 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 [TSputsSQL](#) 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](#), [TSputsSQL](#), [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**

select	character string used to construct query.
dateField	character string used to construct query.
table	character string used to construct query.
where	character string used to construct query.
frequency	character string used to specify frequency of the result (only "daily", "monthly" or "annual" are currently supported).
na.as	value to be used to replace NAs in the returned series.
names	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.
con	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

*Get source information from a data object*

---

**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**

[TSsource](#), [TSget](#), [TSconnect](#), [TSdates](#)

---

TSvintages	<i>Indicate all Vintages at a Connection</i>
------------	--

---

**Description**

Indicate all vintages on the database(s) associated with a TSconnection.

**Usage**

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

**Arguments**

con                    A TSconnection object

**Details**

TSvintages returns the vintage identifiers if available. Otherwise NULL is returned. The result, or subsets of it, can be used as the vintage argument in calls to TSget.

**Value**

A vector of strings indicating vintage identifiers

**See Also**

[TSconnect](#)

# Index

- \*Topic **package**
  - TSdbi-package, 2
- \*Topic **ts**
  - dropTStable, 3
  - TScheckdbi, 4
  - TSconnect, 5
  - TSdates, 6
  - TSdbi-package, 2
  - TSdescription, 7
  - TSfinddb, 9
  - TSget, 10
  - TSput, 11
  - TSquery, 13
  - TSsourceInfo, 14
  - TSvintages, 15
- changeTSrepresentation, 11
- conType-class (TSconnect), 5
- dbConnect, 3–5, 7, 10, 13
- dropTStable, 3
- end.TSdates (TSdates), 6
- logicalId-class (TSput), 11
- print, TSdb-method (TSconnect), 5
- show, logicalId-method (TSconnect), 5
- show, TSdb-method (TSconnect), 5
- show, TSmeta-method (TSconnect), 5
- start.TSdates (TSdates), 6
- tfend, 7
- tfend.TSdates (TSdates), 6
- tfstart, 7
- tfstart.TSdates (TSdates), 6
- tfwindow, 11
- TScheckdbi, 4
- TScheckdbi, ANY-method (TScheckdbi), 4
- TScheckdbi, missing-method (TScheckdbi), 4
- TScon (TSsourceInfo), 14
- TSconnect, 3, 5, 11, 15
- TSconnect, character, character-method (TSconnect), 5
- TSdates, 3–5, 6, 9–11, 13, 15
- TSdates, character, ANY-method (TSdates), 6
- TSdates, character, missing-method (TSdates), 6
- TSdb-class (TSconnect), 5
- TSdbi-package, 5, 7, 10, 11, 13
- TSdbi-package, 2
- TSdbi.Intro (TSdbi-package), 2
- TSdbiMethods, 3
- TSdelete (TSput), 11
- TSdelete, character, ANY, ANY, ANY-method (TSput), 11
- TSdelete, character, ANY-method (TSput), 11
- TSdelete, character, missing, ANY, ANY-method (TSput), 11
- TSdelete, character, missing-method (TSput), 11
- TSdescription, 7, 7
- TSdescription, ANY, missing-method (TSdescription), 7
- TSdescription, character, ANY-method (TSdescription), 7
- TSdescription, character, missing-method (TSdescription), 7
- TSdescription, missing, ANY-method (TSdescription), 7
- TSdescription, missing, missing-method (TSdescription), 7
- TSdescription<- (TSdescription), 7
- TSdoc (TSdescription), 7
- TSdoc, ANY, missing-method

- (TSdescription), 7
- TSdoc, character, ANY-method (TSdescription), 7
- TSdoc, character, missing-method (TSdescription), 7
- TSdoc, missing, ANY-method (TSdescription), 7
- TSdoc, missing, missing-method (TSdescription), 7
- TSdoc<- (TSdescription), 7
- TSexists (TSdescription), 7
- TSexists, default-method (TSdescription), 7
- TSextractionDate (TSsourceInfo), 14
- TSfinddb, 9
- TSget, 3–5, 7, 9, 10, 10, 13–15
- TSget, character, ANY-method (TSget), 10
- TSget, character, missing-method (TSget), 10
- TSid-class (TSconnect), 5
- TSlabel (TSdescription), 7
- TSlabel, ANY, missing-method (TSdescription), 7
- TSlabel, character, ANY-method (TSdescription), 7
- TSlabel, character, missing-method (TSdescription), 7
- TSlabel, missing, ANY-method (TSdescription), 7
- TSlabel, missing, missing-method (TSdescription), 7
- TSlabel<- (TSdescription), 7
- TSmeta (TSdescription), 7
- TSmeta, ANY, missing-method (TSdescription), 7
- TSmeta, character, ANY-method (TSdescription), 7
- TSmeta, character, missing-method (TSdescription), 7
- TSmeta-class (TSdescription), 7
- TSmeta<- (TSdescription), 7
- TSput, 3–5, 7, 9, 10, 11, 11
- TSput, ANY, character, ANY-method (TSput), 11
- TSput, ANY, character, missing-method (TSput), 11
- TSput, ANY, DBIConnection, missing-method (TSput), 11
- TSput, ANY, missing, missing-method (TSput), 11
- TSputSQL, 11, 13
- TSquery, 13
- TSrefperiod (TSdescription), 7
- TSrefperiod, default-method (TSdescription), 7
- TSrefperiod<- (TSdescription), 7
- TSreplace (TSput), 11
- TSreplace, default-method (TSput), 11
- TSseriesIDs (TSsourceInfo), 14
- TSsource, 15
- TSsource (TSdescription), 7
- TSsource, ANY, missing-method (TSdescription), 7
- TSsource, character, ANY-method (TSdescription), 7
- TSsource, character, missing-method (TSdescription), 7
- TSsource, missing, ANY-method (TSdescription), 7
- TSsource, missing, missing-method (TSdescription), 7
- TSsource<- (TSdescription), 7
- TSsourceInfo, 14
- TSvintages, 15
- TSvintages, ANY-method (TSvintages), 15
- TSvintages, missing-method (TSvintages), 15