

# Package ‘TSfame’

April 30, 2015

**Version** 2015.4-1

**Title** 'TSdbi' Extensions for Fame

**Description** A 'fame' interface for 'TSdbi'.

Comprehensive examples of all the 'TS\*' packages is provided in the vignette Guide.pdf with the 'TSdata' package.

**Depends** R (>= 2.8.0), TSdbi (>= 2015.1-1)

**Imports** methods, DBI (>= 0.3.1), tframe, tframePlus, fame (>= 2.12),  
tis

**Suggests** tseries, tfplot, zoo

**BuildVignettes** true

**License** GPL-2

**Copyright** 2007-2011 Bank of Canada. 2012-2015 Paul Gilbert.

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

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

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

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-04-30 20:39:37

## R topics documented:

TSfameMethods . . . . .	2
vintageMap . . . . .	6
<b>Index</b>	<b>7</b>

---

TSfameMethods

*TSdbi fame Methods*


---

## Description

Methods for TSdbi fame time series database connection.

## Usage

```
fameLocal(...)
fameServer(...)
## S4 method for signature 'TSfameLocalConnection,missing'
TSconnect(q, dbname,
accessMode = if (file.exists(dbname)) "shared" else "create",
current = NA, ...)
## S4 method for signature 'TSfameServerConnection,missing'
TSconnect(q, dbname,
service = "", host = "", user = "", password = "", current = NA, ...)

## S4 method for signature 'character,TSfameLocalConnection'
TSget(serIDs, con,
TSrepresentation = getOption("TSrepresentation"),
tf = NULL, start = tfstart(tf), end = tfend(tf), names = NULL,
TSdescription = FALSE, TSdoc = FALSE, TSlabel=FALSE, TSource=TRUE,
vintage=getOption("TSvintage"), ...)
## S4 method for signature 'character,TSfameServerConnection'
TSget(serIDs, con,
TSrepresentation = getOption("TSrepresentation"),
tf = NULL, start = tfstart(tf), end = tfend(tf), names = NULL,
TSdescription = FALSE, TSdoc = FALSE, TSlabel=FALSE, TSource=TRUE,
vintage=getOption("TSvintage"), ...)

## S4 method for signature 'ANY,character,TSfameLocalConnection'
TSput(x, serIDs=seriesNames(x),
con, TSdescription.=TSdescription(x),
TSdoc.=TSdoc(x), TSlabel.=NULL, TSource.=NULL, warn = TRUE, ...)
## S4 method for signature 'ANY,character,TSfameServerConnection'
TSput(x, serIDs=seriesNames(x),
con, TSdescription.=TSdescription(x),
TSdoc.=TSdoc(x), TSlabel.=NULL, TSource.=NULL, warn = TRUE, ...)

## S4 method for signature 'character,TSfameLocalConnection'
TSdates(serIDs, con,
vintage=getOption("TSvintage"), panel=NULL, ...)
## S4 method for signature 'character,TSfameServerConnection'
TSdates(serIDs, con,
vintage=getOption("TSvintage"), panel=NULL, ...)
```

```

    ## S4 method for signature 'character,TSfameLocalConnection'
TSexists(
  serIDs, con, vintage=NULL, panel=NULL, ...)
    ## S4 method for signature 'character,TSfameServerConnection'
TSexists(
  serIDs, con, vintage=NULL, panel=NULL, ...)

    ## S4 method for signature 'character,TSfameLocalConnection'
TDelete(serIDs, con,
        vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
    ## S4 method for signature 'character,TSfameServerConnection'
TDelete(serIDs, con,
        vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)

    ## S4 method for signature 'character,TSfameLocalConnection'
TDescription(x, con, ...)
    ## S4 method for signature 'character,TSfameServerConnection'
TDescription(x, con, ...)

    ## S4 method for signature 'character,TSfameLocalConnection'
TDoc(x, con, ...)
    ## S4 method for signature 'character,TSfameServerConnection'
TDoc(x, con, ...)

    ## S4 method for signature 'character,TSfameLocalConnection'
TLabel(x, con, ...)
    ## S4 method for signature 'character,TSfameServerConnection'
TLabel(x, con, ...)

    ## S4 method for signature 'character,TSfameLocalConnection'
TSource(x, con, ...)
    ## S4 method for signature 'character,TSfameServerConnection'
TSource(x, con, ...)

    ## S4 method for signature 'TSfameLocalConnection'
TVintages(con)
    ## S4 method for signature 'TSfameServerConnection'
TVintages(con)

```

### Arguments

q	A character string indicating the query interface to use, or a database connection object.
dbname	The name of the [server and] database to which the connection should be established, omitted if q is a database connection object.
con	A database connection.
serIDs	identifiers for series to extract.

vintage	character string indicating vintage of the series on the databases.
panel	character string indicating panel of the series on the database (not supported by this database).
x	data series to put on the database, or a series identifier for TSdescription and TSdoc or, for print, a database connection as returned by TSconnect.
TSrepresentation	time series representation to use for the result.
names	Optional character vector to use for series names. Defaults to serIDs unless vintage is a vector, in which case it defaults to vintage.
tf	Optional tframe to use for truncating data. (See tfwindow.)
start	Optional start date to use for truncating data. (See tfwindow.)
end	Optional end date to use for truncating data. (See tfwindow.)
TSdescription	TRUE or FALSE indicating whether description should be extracted
TSdescription.	Description to put on the database.
TSdoc	TRUE or FALSE indicating whether documentation should be extracted.
TSdoc.	Documentation to put on the database.
TSlabel	For compatibility with other TS packages. Not supported in Fame
TSlabel.	For compatibility with other TS packages. Not supported in Fame
TSsource	For compatibility with other TS packages. Not supported in Fame
TSsource.	For compatibility with other TS packages. Not supported in Fame
warn	Logical indicating if warning messages should be suppressed.
accessMode	Optional character indicating the mode to use when opening the Fame database when establishing the first connection. After the first connection, "read" will be used for reading and "update" for writing.
host	A character string indicating a server.
service	A character string indicating a server port number.
user	A character string indicating a user id (if needed).
password	A character string indicating a password (if needed).
current	A character string indicating a vintage that should be considered the current vintage (see details).
...	Arguments passed to other methods.

## Details

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

This package provides a wrapper to the **fame** package to provide a **TSdbi** API for users. The `TSfameLocalConnection` establishes a connection using the locally fame which can access local databases or databases on a remote server, specified by the server name followed by a space and the database name. The `TSfameServerConnection` establishes a connection to a remote server using one of Fame's remote server protocols. See package **fame** for more specific details.

There is a simplistic mechanism for accessing vintages of data stored in different Fame databases. To use this, the `dbname` argument to `TSconnect` should be a vector of character strings indicating the databases, with names of the elements indicating the vintages. The `vintage` argument to `TSget` is then a character vector used to select a subset of `dbname`. The `dbname` argument to `TSconnect` can also be specified to indicate the version that should be considered current. (That is, the user needs to know that information and specify it, rather than having it supplied by the database.) In other respects the vintages support is then similar to other TSdbi packages such as **TSMYSQL**, and examples in those packages may be useful. Some examples are also provided in the **TSfame** Guide vignette.

### Value

depends.

### See Also

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

### Examples

```
require("tfplot")
con <- try(TSconnect(dbDriver("fame"), dbname="test.db"))
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")
  dbDisconnect(con)
}
```

---

vintageMap	<i>Read vintage names and file paths to use for TSconnect</i>
------------	---------------------------------------------------------------

---

**Description**

Read vintage names and file paths to use for TSconnect.

**Usage**

```
vintageMap(file)
```

**Arguments**

file            A character string indicating a file.

**Details**

The argument `file` should be a character string indicating a file that will be read. The file should have lines with two space separated columns, the first indicating a name to be used for a vintage, e.g., 2009-01-07, and the second indicating the path to the database file to use for that vintage, e.g., /path/to/ets\_2009-01-07.db. The file is read and the result is a string that can be passed as the `dbname` to TSfame's TSconnect method.

**Value**

a character vector with names

**See Also**

[TSconnect](#)

# Index

\*Topic **ts**  
    TSfameMethods, [2](#)  
    vintageMap, [6](#)

dbConnect, [5](#)

fameLocal (TSfameMethods), [2](#)  
fameServer (TSfameMethods), [2](#)

getfame, [5](#)

TSconnect, [6](#)  
TSconnect, TSfameLocalConnection, missing-methods  
    (TSfameMethods), [2](#)  
TSconnect, TSfameServerConnection, missing-methods  
    (TSfameMethods), [2](#)

TSdates, character, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSdates, character, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

TSdelete, character, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSdelete, character, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

TSdescription, character, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSdescription, character, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

TSdoc, character, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSdoc, character, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

TSexists, character, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSexists, character, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

TSfameLocalConnection-class  
    (TSfameMethods), [2](#)

TSfameMethods, [2](#)

TSfameServerConnection-class  
    (TSfameMethods), [2](#)

TSget, character, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSget, character, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

TSlabel, character, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSlabel, character, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

TSput, [5](#)  
TSput, ANY, character, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSput, ANY, character, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

TSsource, character, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSsource, character, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

TSvintages, TSfameLocalConnection-method  
    (TSfameMethods), [2](#)  
TSvintages, TSfameServerConnection-method  
    (TSfameMethods), [2](#)

vintageMap, [6](#)