

# Package ‘dsepadi’

October 3, 2006

**Title** Dynamic Systems Estimation - PADI extensions

**Description** Multivariate Time Series - PADI extensions

**Depends** R (>= 2.0.0), tframe, dse1 (>= 2006.10-1)

**Version** 2006.10-1

**Date** 2006-10-03

**LazyLoad** yes

**License** Free. See the LICENCE file for details.

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

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

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

## R topics documented:

TSPADIdata . . . . .	2
TSPADIdata2 . . . . .	3
availability . . . . .	4
freeze.TSPADIdata . . . . .	5
freeze.tfPADIdata . . . . .	6
getpadi . . . . .	7
identifiers . . . . .	9
inputData.TSPADIdata . . . . .	10
modify . . . . .	10
modify.TSPADIdata . . . . .	12
print.TSPADIdata . . . . .	13
print.tfPADIdata . . . . .	14
putpadi . . . . .	15
refresh . . . . .	16
retrieve.and.verify.data . . . . .	16
setTSPADIdata . . . . .	17
settfPADIdata . . . . .	18
sourceInfo . . . . .	19
sourceInfo.TSPADIdata . . . . .	19
sourceInfo.TSdata . . . . .	20

sourcedb . . . . .	21
sourceserver . . . . .	22
tfPADIdata . . . . .	22
tfperiods.tfPADIdata . . . . .	24
tfputpadi . . . . .	24
tsp.TSPADIdata . . . . .	26
tsp.tfPADIdata . . . . .	26

<b>Index</b>	<b>27</b>
--------------	-----------

---

TSPADIdata	<i>Constructor for TSPADIdata object</i>
------------	--

---

## Description

Make a TSPADIdata object from identifier information

## Usage

```
TSPADIdata(output=NULL,          input=NULL,
            output.server=server, input.server=server,
            output.db=db,         input.db=db,
            output.transforms="", input.transforms="",
            output.names=NULL,    input.names=NULL,
            start=NA, end=NA, frequency=NA,
            pad=FALSE, pad.start=pad, pad.end=pad,
            server="", db="", start.server=NULL,
            server.process=NULL, cleanup.script=NULL,
            stop.on.error=TRUE, warn=TRUE)
```

```
is.TSPADIdata(obj)
```

## Arguments

output	see tfPADIdata description of series.
output.server	see tfPADIdata descriptions.
output.db	see tfPADIdata descriptions.
output.transforms	see tfPADIdata descriptions.
output.names	see tfPADIdata descriptions.
input	see tfPADIdata description of series.
input.server	see tfPADIdata descriptions.
input.db	see tfPADIdata descriptions.
input.transforms	see tfPADIdata descriptions.
input.names	see tfPADIdata descriptions.
start	see tfPADIdata descriptions.
end	see tfPADIdata descriptions.

frequency	see tfPADIdata descriptions.
pad	see tfPADIdata descriptions.
pad.start	see tfPADIdata descriptions.
pad.end	see tfPADIdata descriptions.
server	see tfPADIdata descriptions.
db	see tfPADIdata descriptions.
start.server	see tfPADIdata descriptions.
server.process	see tfPADIdata descriptions.
cleanup.script	see tfPADIdata descriptions.
stop.on.error	see tfPADIdata descriptions.
warn	see tfPADIdata descriptions.
obj	any object.

### Details

This is the way to produce an object which can be used for PADI database access. Input and output values are passed to tfPADIdata so the TSdata object return is a list of tfPADIdata objects called input and output (but either of these can be NULL).

### Value

An object of class "TSPADIdata" "TSdata".

### See Also

[freeze.TSPADIdata](#) [freeze.TSPADIdata](#) [tfPADIdata](#) [TSPADIdata2](#) [setTSPADIdata](#)

### Examples

```
z <- TSPADIdata(output=c("seriesA", "seriesB"), server="myserver")
is.TSPADIdata(z)
```

---

TSPADIdata2	<i>Constructor for TSPADIdata object</i>
-------------	--

---

### Description

Make a TSPADIdata object from identifier information

### Usage

```
TSPADIdata2(input=NULL, output=NULL,
  start = NA, end = NA, frequency = NA,
  pad.start = FALSE, pad.end = FALSE,
  start.server = NULL, server.process = NULL, cleanup.script = NULL,
  stop.on.error =TRUE, warn =TRUE)
```

## Arguments

<code>input</code>	Lists of character vectors indicating data to be used as input. Each vector has 5 elements indicating the server, the database, the series identifier, any transformation, and a name.
<code>output</code>	List (as for input) indicating outputs. At least one of input or output should be specified.
<code>start, end, frequency, pad.start, pad.end</code>	Information about data retrieval. See <code>tfPADIdata</code> for more details.
<code>start.server, server.process, cleanup.script, stop.on.error, warn</code>	Additional information server operation. See <code>tfPADIdata</code> for more details.

## Details

This is an alternate way to produce an object which can be used for PADI database access. Input and output values are passed to `tfPADIdata` so the `TSdata` object return is a list of `tfPADIdata` objects called input and output (but either of these can be `NULL`).

## Value

An object of class "TSPADIdata" "TSdata".

## See Also

[freeze.TSPADIdata](#) [freeze.TSPADIdata](#) [tfPADIdata](#) [TSPADIdata2](#) [setTSPADIdata](#)

## Examples

```
inflation.sa.names <- TSPADIdata2(
  output = list(
    c("ets","", "ippia401", "ytoypc", "industrial product price index"),
    c("ets","", "b820600", "ytoypc", "cpi"),
    c("ets","", "b820655", "ytoypc", "cpixfe"),
    c("ets","", "b820672", "ytoypc", "price goods excl food & energy"),
    c("ets","", "b820678", "ytoypc", "price services"),
    c("ets","", "b800938", "ytoypc", "total unit labour costs")
  ), stop.on.error=TRUE, warn=TRUE )
```

---

availability

*Check Data Availability*

---

## Description

Check the dates for which date is available.

## Usage

```
availability(obj, ...)
## Default S3 method:
availability(obj, names=NULL, server="ets", dbname="",
             verbose=TRUE, timeout=60, stop.on.error=TRUE, warn=TRUE, ...)
## S3 method for class 'tfPADIdata':
availability(obj, verbose=TRUE, timeout=60, ...)
```

**Arguments**

<code>obj</code>	a vector of series identifiers or an object of class <code>tfPADIdata</code> . See <code>TSdata.TSPADIdata</code> .
<code>...</code>	arguments passed to other methods.
<code>verbose</code>	TRUE or FALSE indicating if the results should be printed.
<code>timeout</code>	an integer indicating the number of seconds to wait before concluding that the server is not available.
<code>names</code>	A character vector of names to be associated with the ids in printed results.
<code>server</code>	A character vector of servers to be associated with the ids.
<code>dbname</code>	A character vector of databases to be associated with the ids.
<code>stop.on.error</code>	logical indicating if the function should stop if any series produces an error, or continue with other series.
<code>warn</code>	logical indicating if warning messages should be suppressed.

**Details**

If `verbose` is TRUE then the start, end, and frequency are printed for each series in `data.id`. A list is returned (invisibly) with the same information. The default method works for a character vector argument. An argument of class `TSPADIdata` may supply some additional information, and several of the indicated optional arguments do not apply to that method.

**Value**

A list with elements `start`, `end`, and `frequency`.

**See Also**

`tfPADIdata`, `TSPADIdata`, `setTSPADIdata`, `retrieve.and.verify.data`

**Examples**

```
if(require("padi") && checkPADIServer("ets")) {
  library() # debugging to see where padi is found if this fails
  d <- tfPADIdata("D1", server="ets")
  availability(d)}
```

---

`freeze.TSPADIdata`    *Get fixed data snapshot using TSPADI*

---

**Description**

Get a `TSdata` structure from TSPADI Database Interface

**Usage**

```
## S3 method for class 'TSPADIdata':
freeze(data, timeout=60, ...)
```

**Arguments**

data	An object of class TSPADIData.
timeout	an integer indicating the number of seconds to wait before concluding that the server is not available.
...	Arguments passed to other methods.

**Details**

This function extracts data using the TS PADI interface to a database. See TSPADIData, tfPADIData, and freeze for more details.

**Value**

A TSdata object taken from the database.

**See Also**

[freeze TSdata TSPADIData](#)

**Examples**

```
inflation.sa.names <- TSPADIData2(
  output = list(
    c("ets","", "b820678", "ytoypc", "price services"),
    c("ets","", "b800938", "ytoypc", "total unit labour costs")
  ), stop.on.error=TRUE, warn=TRUE )
if(require("padi") & require("dse2") && checkPADIServer("ets"))
  z <- freeze(inflation.sa.names)
```

---

`freeze.tfPADIData`    *Get fixed data snapshot*

---

**Description**

Get a time series matrix structure from a database

**Usage**

```
## S3 method for class 'tfPADIData':
freeze(data, timeout=60, ...)
## S3 method for class 'FAMEData':
freeze(data, ...)
```

**Arguments**

data	A description of how to get data.
timeout	an integer indicating the number of seconds to wait before concluding that the server is not available.
...	Arguments passed to other methods.

## Details

This function extracts data from a database (for example using the TS PADI programs are available at [www.bank-banque-canada.ca/pgilbert](http://www.bank-banque-canada.ca/pgilbert) ). This method is generic. Typically the argument data is a tfPADIdata or TSPADIdata object identifying the source of the data. See help for tfPADIdata and TSPADIdata. The default method usually just returns its argument, so freeze has no effect. This way freeze can be used to write functions which will take a snapshot from the database when they execute or will work with an already fixed copy of data if that is what is supplied. The default does allow for a character argument, in which case it is used to construct a tfPADIdata object using server="ets", then freeze that object. This allows for a simple syntax to grab a series from the database. The server="ets" is for convenience at the Bank of Canada and another default server might be more convenient elsewhere.

## Value

A time series matrix.

## See Also

[tfPADIdata freeze.TSPADIdata TSPADIdata](#)

## Examples

```
if (require("padi") && checkPADIServer("ets")){
  z1 <- freeze("D1")
  z2 <- freeze(tfPADIdata(c("B1630", "B1642"), server="ets"))
}
```

---

getpadi

*Get Data from TSPADI Database Interface*


---

## Description

Get data from a TSPADI database interface.

## Usage

```
getpadi(series, server=Sys.info()[["nodename"]], dbname="",
        start.server=TRUE, server.process=PADIServerProcess(),
        cleanup.script=PADICleanupScript(),
        starty=0,startm=0,startd=1, endy=0,endm=0,endd=1,
        nobs=0,max.obs=2000, transformations=NULL, pad=FALSE,
        user=Sys.info()[["user"]], passwd="",
        stop.on.error=TRUE, use.tframe=FALSE, warn=TRUE, timeout=60)
```

**Arguments**

<code>series</code>	A character string giving the name of the series. Alternately, series can be a vector of character strings specifying multiple series.
<code>server</code>	A character string giving the network name of the server which is to be requested to supply the series. If series specifies multiple series and they are not all on the same server then server should be a vector of character strings with elements corresponding to the elements of series.
<code>dbname</code>	A character string giving additional information to the server about the location of the series (eg. the name of a database). If series specifies multiple series and they are not all on the same database then dbname should be a vector of character strings with elements corresponding to the elements of series.
<code>starty</code>	An integer indicating the starting year.
<code>startm</code>	An integer indicating the starting period.
<code>startd</code>	An integer indicating the starting day.
<code>endy</code>	An integer indicating the ending year.
<code>endm</code>	An integer indicating the ending period.
<code>endd</code>	An integer indicating the ending day.
<code>nobs</code>	The number of observations.
<code>max.obs</code>	integer indicating the possible returned data size used to define the size of the buffer to prepare.
<code>transformations</code>	A character string giving transformations to be applied to the series (e.g. "log"). If multiple series are being requested then transformations can be a single string, in which case it is applied to all series, or a vector of character strings, one for each series. If no transformation is to be applied to some series then "" should be used.
<code>pad</code>	If FALSE (default) then all series are truncated to the interection of available time periods (i.e. the latest start date and earliest end date). If TRUE then series are padded with NA so the result starts at the earliest available observation and ends at the last available observation.
<code>start.server</code>	try to start a server if one is not running
<code>server.process</code>	command to execute in an attempt to start a server
<code>cleanup.script</code>	command to execute to terminate a server if one is started
<code>user</code>	user id for access to the database (if necessary)
<code>passwd</code>	password for access to the database (if necessary)
<code>stop.on.error</code>	If TRUE then stop is executed when an error occurs. Otherwise, the error message is returned and the calling program must deal with it.
<code>use.tframe</code>	If use.tframe=FALSE then ts() is used to construct the time series, otherwise the tframe utilities are used. Certain transformations available with DSE require the tframe stucture and an error may result if these transformations are attempted with use.tframe=FALSE.
<code>warn</code>	Print warning messages for some crude frequency conversions (weekly data).
<code>timeout</code>	an integer indicating the number of seconds to wait before concluding that the server is not available.



## Details

See `getpadi.default`

## Value

A time series matrix with a column for each series.

## See Also

[putpadi](#)

---

identifiers	<i>Get Identifiers from a data object</i>
-------------	---

---

## Description

Get Identifiers from a data object

## Usage

```
identifiers(obj)
## Default S3 method:
identifiers(obj)
## S3 method for class 'tfPADIdata':
identifiers(obj)
```

## Arguments

`obj` An object which contains source series identifier information.

## Value

A list with elements input and output which are strings indicating the input and output series identifiers.

## See Also

[tfPADIdata](#) [TSPADIdata](#) [sourceInfo](#) [sourcedb](#)

## Examples

```
if(require("padi") && checkPADIservers("ets")) {
  d <- tfPADIdata("D1", server="ets")
  identifiers(d)
}
```

---



*TS Input and Output Specific Methods*


---

### Description

See the generic function description.

### Usage

```
## S3 method for class 'TSPADIdata':
inputData(x, series=seq(length=nseriesInput(x)))
## S3 method for class 'TSPADIdata':
outputData(x, series=seq(length=nseriesOutput(x)))
## S3 method for class 'TSPADIdata':
periodsInput(x)
## S3 method for class 'TSPADIdata':
periodsOutput(x)
```

### Arguments

<code>x</code>	a TSPADIdata object.
<code>series</code>	series to select. Passed to selectSeries.

---

modify

*modify a database descriptor object*


---

### Description

Modify a database descriptor object with new information

### Usage

```
modify(obj, start=NA, end=NA, frequency=NA,
       pad=NA, pad.start=pad, pad.end=pad,
       server=NA, db=NA, start.server=NA,
       server.process=NA, cleanup.script=NA,
       stop.on.error=NA, warn=NA,
       append=NA, use.tframe=NA, ...)
## S3 method for class 'tfPADIdata':
modify(obj,
       start=NA, end=NA, frequency=NA,
       pad=NA, pad.start=pad, pad.end=pad,
       server=NA, db=NA, start.server=NA,
       server.process=NA, cleanup.script=NA,
       stop.on.error=NA, warn=NA,
       append=NA, use.tframe=NA,
       series=NA, transforms=NA, names=NA, ...)
```

**Arguments**

<code>obj</code>	database descriptor object to be modified.
<code>append</code>	if specified, append should be a list of (series, server, db, transforms, names) as specified for <code>tfPADIdata</code> . These are appended as additional series to the object.
<code>series</code>	see <code>tfPADIdata</code> descriptions.
<code>server</code>	see <code>tfPADIdata</code> descriptions.
<code>db</code>	see <code>tfPADIdata</code> descriptions.
<code>transforms</code>	see <code>tfPADIdata</code> descriptions.
<code>start</code>	see <code>tfPADIdata</code> descriptions.
<code>end</code>	see <code>tfPADIdata</code> descriptions.
<code>frequency</code>	see <code>tfPADIdata</code> descriptions.
<code>names</code>	see <code>tfPADIdata</code> descriptions.
<code>pad</code>	see <code>tfPADIdata</code> descriptions.
<code>pad.start</code>	see <code>tfPADIdata</code> descriptions.
<code>pad.end</code>	see <code>tfPADIdata</code> descriptions.
<code>use.tframe</code>	see <code>tfPADIdata</code> descriptions.
<code>start.server</code>	see <code>tfPADIdata</code> descriptions.
<code>server.process</code>	see <code>tfPADIdata</code> descriptions.
<code>cleanup.script</code>	see <code>tfPADIdata</code> descriptions.
<code>stop.on.error</code>	see <code>tfPADIdata</code> descriptions.
<code>warn</code>	see <code>tfPADIdata</code> descriptions.
<code>...</code>	arguments to be passed to other methods.

**Details**

Any specified (non NA) optional arguments are used to modify the object. Values replace existing values (except in the case of `append`).

**Value**

A object of class "tfPADIdata" which can be used to retrieve a matrix time series object.

**See Also**

[tfPADIdata freeze modify.TSPADIdata](#)

**Examples**

```
if(require("padi")) {
  z <- tfPADIdata( c("seriesA", "seriesB"), server="myserver")
  z <- modify(z, pad.end=TRUE )
}
```

---

modify.TSPADIdata    *Modify a TSPADIdata Object*

---

## Description

Modify a TSPADIdata object with new information

## Usage

```
## S3 method for class 'TSPADIdata':
modify(obj,
      start=NA, end=NA, frequency=NA,
      pad=NA, pad.start=pad, pad.end=pad,
      server=NA, db=NA, start.server=NA,
      server.process=NA, cleanup.script=NA,
      stop.on.error=NA, warn=NA,
      append=NA, use.tframe=NA,
      output=NA, input=NA,
      output.server=NA, input.server=NA,
      output.db=NA, input.db=NA,
      output.transforms=NA, input.transforms=NA,
      output.names=NA, input.names=NA,
      ...)
```

## Arguments

obj	a tfPADIdata object.
output	see tfPADIdata description of series.
output.server	see tfPADIdata descriptions.
output.db	see tfPADIdata descriptions.
output.transforms	see tfPADIdata descriptions.
output.names	see tfPADIdata descriptions.
input	see tfPADIdata descriptions of series
input.server	see tfPADIdata descriptions.
input.db	see tfPADIdata descriptions.
input.transforms	see tfPADIdata descriptions.
input.names	see tfPADIdata descriptions.
start	see tfPADIdata descriptions.
end	see tfPADIdata descriptions.
frequency	see tfPADIdata descriptions.
pad	see tfPADIdata descriptions.
pad.start	see tfPADIdata descriptions.
pad.end	see tfPADIdata descriptions.

```

start.server see tfPADIdata descriptions.
server.process
               see tfPADIdata descriptions.
server        see tfPADIdata descriptions.
db            see tfPADIdata descriptions.
cleanup.script
               see tfPADIdata descriptions.
stop.on.error
               see tfPADIdata descriptions.
warn          see tfPADIdata descriptions.
append        see tfPADIdata descriptions.
use.tframe    see tfPADIdata descriptions.
...           arguments to be passed to other methods.

```

### Details

This is the way to modify a TSPADIdata object.

### Value

An object of class "TSPADIdata" "TSdata".

### See Also

[TSPADIdata](#) [modify.tfPADIdata](#) [freeze.TSPADIdata](#) [freeze.TSPADIdata](#) [TSPADIdata](#)

### Examples

```

inflation.sa.names <- TSPADIdata2(
  output = list(
    c("ets","", "b820678", "ytoypc", "price services"),
    c("ets","", "b800938", "ytoypc", "total unit labour costs")
  ), stop.on.error=TRUE, warn=TRUE )
z <- modify(inflation.sa.names, server="newserver")

```

---

print.TSPADIdata     *Print Specific Methods*

---

### Description

See the generic function description.

### Usage

```

## S3 method for class 'TSPADIdata':
print(x, ...)

```

**Arguments**

`x` a TSPADIdata object.  
`...` arguments to be passed to other methods.

**See Also**

[print.summary](#)

---

`print.tfPADIdata` *Print a tfPADIdata Object*

---

**Description**

See the generic function description.

**Usage**

```
## S3 method for class 'tfPADIdata':  
print(x, ...)
```

**Arguments**

`x` a tfPADIdata object.  
`...` arguments to be passed to other methods.

**Side Effects**

The object is printed.

**See Also**

[print.tfPADIdata](#)

**Examples**

```
if(require("padi") && checkPADIServer("ets"))  
  print(tfPADIdata(c("series1", "series2"), server="ets"))
```

putpadi

*PADI Functions***Description**

see details

**Usage**

```

putpadi(data, server=Sys.info()[["nodename"]], dbname="",
        series=seriesNames(data),
        start.server=TRUE, server.process=PADIserverProcess(),
        cleanup.script=PADICleanupScript(),
        user=Sys.info()[["user"]], passwd="",
        stop.on.error=TRUE, warn=TRUE, timeout=60)
## S3 method for class 'TSdata':
putpadi(data, server=Sys.info()[["nodename"]], dbname,
        series=seriesNames(data),
        start.server=TRUE, server.process=padi.server.process(),
        cleanup.script=padi.cleanup.script(),
        user=Sys.info()[["user"]], passwd="",
        stop.on.error=TRUE, warn=TRUE, timeout=60)

```

**Arguments**

<code>data</code>	a <code>tfPADIdata</code> object with data to put on the server.
<code>server</code>	string or vector of strings indicating the server where the series will be found. Scalar values are expanded to a vector of appropriate length.
<code>dbname</code>	string or vector of strings indicating additional information for the server. Scalar values are expanded to a vector of appropriate length.
<code>series</code>	vector of strings to use for series identifiers.
<code>start.server</code>	logical indicating if a (local) server should be started.
<code>server.process</code>	string indicating the name to be used to start a server process.
<code>cleanup.script</code>	string indicating the name to be used to shut down a server process.
<code>user</code>	an optional string used by the server to check permission.
<code>passwd</code>	an optional string used by the server to check permission.
<code>stop.on.error</code>	logical indicating if the function should stop if any series produces an error, or continue with other series.
<code>warn</code>	logical indicating if warning messages should be suppressed.
<code>timeout</code>	an integer indicating the number of seconds to wait before concluding that the server is not available.

**Details**See `putpadi.default`.

**Value**

depends

**See Also**

[getpadi](#) [putpadi](#) [TSdata](#)

---

refresh

*Retrieve new data from a database*

---

**Description**

Extract source information and retrieve an updated version of the data from a database.

**Usage**

```
refresh(data)
```

**Arguments**

**data** An object which has previously been obtained with `freeze`, so it contains source information which can be extracted and `freeze` re-applied.

**Value**

A time series object with class depending on the source information.

**Examples**

```
if(require("padi") && checkPADIServer("ets")) {
  d <- tfPADIData("D1", server="ets")
  d <- freeze(d)
  d <- refresh(d) }
```

---

retrieve.and.verify.data

*Retrieve and Verify Data*

---

**Description**

Retrieve data from a database and verify a subset.

**Usage**

```
retrieve.and.verify.data(data.names,
  verification.data=verification.data, fuzz=1e-10)
```



**Arguments**

`data.names`     An object of class TSdata indicating databases and series.  
`verification.data`  
                   An object of class TSdata.  
`fuzz`            A (small) real number indicating the size of errors to tolerate.

**Details**

The data from the data base is compared with `verification.data`, which would normally be a subset (e.g. a subset used for model estimation). The comparison is done on the sub-sample for which verification data is available. This provides a mechanism for detecting large data revisions or re-basing.

**Value**

An object of class TSdata containing data from the data bases is returned.

**Side Effects**

If errors larger than `fuzz` are found then some warning messages are printed and the series in question can be plotted.

**See Also**

[TSPADIdata](#) [setTSPADIdata](#) [settfPADIdata](#) [freeze](#)

**Examples**

```
if(require("padi") & require("dse2") && checkPADIServer("ets")) {
  dnames <- TSPADIdata(output="D1", server="ets")
  d <- freeze(dnames)
  retrieve.and.verify.data(dnames, verification.data=d)}
```

---

setTSPADIdata	<i>Prompt for Series Names</i>
---------------	--------------------------------

---

**Description**

Prompt for data series names and locations and create a data object.

**Usage**

```
setTSPADIdata( )
```

**Details**

The function prompts for the names and database locations of series to be used for input (exogenous) variables and output (endogenous) variables and creates a data object. The data can be retrieved, in which case it is fixed as currently available (and becomes an object of class 'TSdata'), or it may be left dynamic, in which case it is retrieved whenever it is referenced (and is of class 'TSPADIdata' 'TSdata').

**Value**

An object of class 'TSdata' or of class 'TSPADIdata' 'TSdata'

**See Also**

[freeze TSdata](#)

**Examples**

```
## Not run: data <- setTSPADIdata()
```

---

settfPADIdata	<i>Prompt for series names</i>
---------------	--------------------------------

---

**Description**

Prompt for data series names and locations and create a data object.

**Usage**

```
settfPADIdata(preamble=TRUE)
```

**Arguments**

`preamble` a logical indicating if the description of how to enter data should be printed.

**Details**

The function prompts for the names and database locations of series. The data can be retrieved, in which case it is returned as currently available on the database (and becomes a time series matrix), or it may be left dynamic, in which case it is of class 'tfPADIdata' and the discription can be used anytime to retrieve data.

**Value**

An object of class 'tfPADIdata' (or retrieved time series matrix).

**See Also**

[freeze TSdata](#) [TSdata.object](#)

**Examples**

```
data <- settfPADIdata()
```

---

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

---

## Description

Get source information from an object

## Usage

```
sourceInfo(obj)
## Default S3 method:
sourceInfo(obj)
## S3 method for class 'tfPADIdata':
sourceInfo(obj)
```

## Arguments

`obj` An object which contains source series information.

## Value

A list with elements input and output which are strings indicating the input and output source.

## See Also

[TSPADIdata identifiers](#)

## Examples

```
if(require("padi") && checkPADIServer("ets")) {
  d <- tfPADIdata("D1", server="ets")
  d <- freeze(d)
  sourceInfo(d)}
```

---

sourceInfo.TSPADIdata	<i>TSPADIdata Specific Methods</i>
-----------------------	------------------------------------

---

## Description

See the generic function description.

**Usage**

```
## S3 method for class 'TSPADIdata':
sourceInfo(obj)
## S3 method for class 'TSPADIdata':
sourcedb(obj)
## S3 method for class 'TSPADIdata':
sourceserver(obj)
## S3 method for class 'TSPADIdata':
availability(obj, verbose=TRUE, timeout=60, ...)
## S3 method for class 'TSPADIdata':
identifiers(obj)
## S3 method for class 'TSPADIdata':
periods(x)
```

**Arguments**

obj	a TSPADIdata object.
x	a TSPADIdata object.
verbose	a logical indicating if extra information should be printed.
timeout	an integer indicating the number of seconds to wait before concluding that the server is not available.
...	(further arguments, currently disregarded).

**See Also**

[sourceInfo](#)

---

sourceInfo.TSdata    *tfPADI Specific Methods*

---

**Description**

See the generic function description.

**Usage**

```
## S3 method for class 'TSdata':
sourceInfo(obj)
## S3 method for class 'TSestModel':
sourceInfo(obj)

## S3 method for class 'TSdata':
sourcedb(obj)
## S3 method for class 'TSestModel':
sourcedb(obj)

## S3 method for class 'TSdata':
sourceserver(obj)
## S3 method for class 'TSestModel':
sourceserver(obj)
```

```
## S3 method for class 'TSdata':
identifiers(obj)
## S3 method for class 'TSestModel':
identifiers(obj)
```

### Arguments

`obj`                      An object of class `TSdata` or `TSestModel`.

### See Also

[sourceInfo](#)

---

sourcedb	<i>Get sourcedb from a data object</i>
----------	--

---

### Description

Get sourcedb from a data object

### Usage

```
sourcedb(obj)
## Default S3 method:
sourcedb(obj)
## S3 method for class 'tfPADIdata':
sourcedb(obj)
```

### Arguments

`obj`                      An object which contains source data base information.

### Value

A list with elements input and output which are strings indicating the input and output sourcedb.

### See Also

[TSPADIdata](#) [identifiers](#) [sourceInfo](#)

### Examples

```
if(require("padi") && checkPADIServer("ets")) {
  d <- tfPADIdata("D1", server="ets")
  d <- freeze(d)
  sourcedb(d) }
```

---

sourceserver	<i>Get source server from a data object</i>
--------------	---

---

### Description

Get source server information from an object

### Usage

```
sourceserver(obj)
## Default S3 method:
sourceserver(obj)
## S3 method for class 'tfPADIdata':
sourceserver(obj)
```

### Arguments

obj                      An object which contains source series information.

### Value

A list with elements input and output which are strings indicating the input and output sourceserver.

### See Also

[TSPADIdata](#) [identifiers](#) [sourcedb](#) [sourceInfo](#)

### Examples

```
if(require("padi") && checkPADIServer("ets")) {
  d <- tfPADIdata("D1", server="ets")
  sourceserver(d)}
```

---

tfPADIdata	<i>make tfPADIdata object</i>
------------	-------------------------------

---

### Description

Make a tfPADIdata object from a vector of identifiers

### Usage

```
tfPADIdata(series, server = "", db = "", transforms = "",
  start=NA, end=NA, frequency=NA, names=NULL,
  pad=FALSE, pad.start=pad, pad.end=pad,
  use.tframe=TRUE,
  start.server=FALSE,
  server.process=PADIServerProcess(),
  cleanup.script=PADICleanupScript(),
  stop.on.error=TRUE, warn=TRUE)
is.tfPADIdata(obj)
```

**Arguments**

<code>series</code>	vectors of strings indicating series identifiers.
<code>server</code>	string or vector of strings indicating the server where the series will be found. Scalar values are expanded to a vector of appropriate length.
<code>db</code>	string or vector of strings indicating additional information for the server. Scalar values are expanded to a vector of appropriate length.
<code>transforms</code>	see details.
<code>start</code>	a two element vector of starting year and period which is used when requesting data.
<code>end</code>	a two element vector of end year and end period which is used when requesting data.
<code>frequency</code>	a scalar indicating frequency. This is not used in requesting data, but if supplied it is used as a check of returned data. A warning is issued if returned data does not have this frequency.
<code>names</code>	names to replace series identifiers.
<code>pad</code>	TRUE or FALSE indicating if data should be padded with NAs on both ends to the length of the longest available series. If padding is not done then series are truncated to the intersection of available data so there are no NAs in the result. <code>pad.start</code> and <code>pad.end</code> are the arguments which are actually used. <code>pad</code> is just a convenient way to specify both.
<code>pad.start</code>	logical indicating if NAs should be padded to beginning of data.
<code>pad.end</code>	logical indicating if NAs should be padded to end of data.
<code>use.tframe</code>	logical indicating if the tframe library methods should be used for the time dimension.
<code>start.server</code>	logical indicating if a (local) server should be started.
<code>server.process</code>	string indicating the name to be used to start a server process.
<code>cleanup.script</code>	string indicating the name to be used to shut down a server process.
<code>stop.on.error</code>	logical indicating if the function should stop if any series produces an error, or continue with other series.
<code>warn</code>	logical indicating if warning messages should be suppressed.
<code>obj</code>	Any object.

**Details**

This is the way to produce an object which can be used for database access. The transforms a vector of strings, one for each series, giving any transformations which should be applied when the data is retrieved (with `eval(call())`, e.g. "log", or "" if no transform is to be applied).

**Value**

A object of class "tfPADIdata" which can be used to retrieve a matrix time series object.

**See Also**

[freeze](#)

**Examples**

```
if(require("padi")) {
  z <- tfPADIdata( c("seriesA", "seriesB"), server="myserver")
  is.tfPADIdata(z)
}
```

---

tfperiods.tfPADIdata

*Specific Methods for Tframed data*

---

**Description**

See the generic function description.

**Usage**

```
## S3 method for class 'tfPADIdata':
tframe(x)
## S3 method for class 'tfPADIdata':
tfperiods(x)
## S3 method for class 'tfPADIdata':
tfstart(x, ...)
## S3 method for class 'tfPADIdata':
tfend(x, ...)
## S3 method for class 'tfPADIdata':
tffrequency(x, ...)
## S3 method for class 'tfPADIdata':
seriesNames(x)
```

**Arguments**

**x** a tfPADIdata object.

**...** (further arguments, currently disregarded).

**See Also**

[tframe](#), [tfperiods](#), [tfstart](#), [tfend](#), [tffrequency](#), [freeze](#), [seriesNames](#)

---

tfputpadi

*Write Data to a Data Server*

---

**Description**

Write data to a server.



**Usage**

```
tfputpadi(data,
  server = Sys.info()[["nodename"]],
  dbname = "",
  series = seriesNames(data),
  start.server = TRUE,
  server.process = padi.server.process(),
  cleanup.script = padi.cleanup.script(),
  user = Sys.info()[["user"]], passwd = "",
  stop.on.error = TRUE, warn = TRUE, timeout=60)
```

**Arguments**

<code>data</code>	a tfPADIdata object with data to put on the server.
<code>server</code>	string or vector of strings indicating the server where the series will be found. Scalar values are expanded to a vector of appropriate length.
<code>dbname</code>	string or vector of strings indicating additional information for the server. Scalar values are expanded to a vector of appropriate length.
<code>series</code>	vector of strings to use for series identifiers.
<code>start.server</code>	logical indicating if a (local) server should be started.
<code>server.process</code>	string indicating the name to be used to start a server process.
<code>cleanup.script</code>	string indicating the name to be used to shut down a server process.
<code>user</code>	an optional string used by the server to check permission.
<code>passwd</code>	an optional string used by the server to check permission.
<code>stop.on.error</code>	logical indicating if the function should stop if any series produces an error, or continue with other series.
<code>warn</code>	logical indicating if warning messages should be suppressed.
<code>timeout</code>	an integer indicating the number of seconds to wait before concluding that the server is not available.

**Details**

This function writes data to a database using the TS PADI programs available at [www.bank-banque-canada.ca/pgilbert](http://www.bank-banque-canada.ca/pgilbert).

**Value**

A tfPADIdata object (as would be suitable for retrieving the data).

**See Also**

[tfPADIdata](#) [putpadi](#)

---

tsp.TSPADIdata	<i>Specific Methods for tsp</i>
----------------	---------------------------------

---

**Description**

See the generic function description.

**Usage**

```
## S3 method for class 'TSPADIdata':  
tsp(x)
```

**Arguments**

x                      a TSPADIdata object.

**See Also**

[tsp](#)

---

tsp.tfPADIdata	<i>Specific Methods for tsp</i>
----------------	---------------------------------

---

**Description**

See the generic function description.

**Usage**

```
## S3 method for class 'tfPADIdata':  
tsp(x)
```

**Arguments**

x                      A tfPADIdata object.

**See Also**

[tsp](#)

# Index

## \*Topic ts

- availability, 4
- freeze.tfPADIdata, 6
- freeze.TSPADIdata, 5
- getpadi, 7
- identifiers, 8
- inputData.TSPADIdata, 9
- modify, 10
- modify.TSPADIdata, 11
- print.tfPADIdata, 13
- print.TSPADIdata, 13
- putpadi, 14
- refresh, 15
- retrieve.and.verify.data, 16
- settfPADIdata, 17
- setTSPADIdata, 17
- sourcedb, 20
- sourceInfo, 18
- sourceInfo.TSdata, 19
- sourceInfo.TSPADIdata, 19
- sourceserver, 21
- tfPADIdata, 22
- tfperiods.tfPADIdata, 23
- tfputpadi, 24
- tsp.tfPADIdata, 25
- tsp.TSPADIdata, 25
- TSPADIdata, 1
- TSPADIdata2, 3
- [.tfPADIdata (tfPADIdata), 22
- availability, 4
- availability.TSPADIdata  
(sourceInfo.TSPADIdata), 19
- freeze, 5, 11, 16–18, 23, 24
- freeze.FAMEData  
(freeze.tfPADIdata), 6
- freeze.tfPADIdata, 6
- freeze.TSPADIdata, 2, 3, 5, 6, 12
- getpadi, 7, 15
- identifiers, 8, 18, 21
- identifiers.TSdata  
(sourceInfo.TSdata), 19
- identifiers.TSestModel  
(sourceInfo.TSdata), 19
- identifiers.TSPADIdata  
(sourceInfo.TSPADIdata), 19
- inputData.TSPADIdata, 9
- is.tfPADIdata (tfPADIdata), 22
- is.TSPADIdata (TSPADIdata), 1
- modify, 10
- modify.tfPADIdata, 12
- modify.TSPADIdata, 11, 11
- outputData.TSPADIdata  
(inputData.TSPADIdata), 9
- periods.TSPADIdata  
(sourceInfo.TSPADIdata), 19
- periodsInput.TSPADIdata  
(inputData.TSPADIdata), 9
- periodsOutput.TSPADIdata  
(inputData.TSPADIdata), 9
- print, 13, 14
- print.tfPADIdata, 13
- print.TSPADIdata, 13
- putpadi, 8, 14, 25
- putpadi.TSdata, 15
- refresh, 15
- retrieve.and.verify.data, 4, 16
- seriesNames, 24
- seriesNames.tfPADIdata  
(tfperiods.tfPADIdata), 23
- settfPADIdata, 16, 17
- setTSPADIdata, 2–4, 16, 17
- sourcedb, 9, 20, 21
- sourcedb.TSdata  
(sourceInfo.TSdata), 19
- sourcedb.TSestModel  
(sourceInfo.TSdata), 19
- sourcedb.TSPADIdata  
(sourceInfo.TSPADIdata), 19
- sourceInfo, 9, 18, 19–21
- sourceInfo.TSdata, 19

- sourceInfo.TSestModel
  - (sourceInfo.TSdata), 19
- sourceInfo.TSPADIdata, 19
- sourceserver, 21
- sourceserver.TSdata
  - (sourceInfo.TSdata), 19
- sourceserver.TSestModel
  - (sourceInfo.TSdata), 19
- sourceserver.TSPADIdata
  - (sourceInfo.TSPADIdata), 19
- summary, 13
  
- tfend, 24
- tfend.tfpADIdata
  - (tfperiods.tfpADIdata), 23
- tffrequency, 24
- tffrequency.tfpADIdata
  - (tfperiods.tfpADIdata), 23
- tfPADIdata, 2–4, 6, 9, 11, 14, 22, 25
- tfperiods, 24
- tfperiods.tfpADIdata, 23
- tfputpadi, 24
- tframe, 24
- tframe.tfpADIdata
  - (tfperiods.tfpADIdata), 23
- tfstart, 24
- tfstart.tfpADIdata
  - (tfperiods.tfpADIdata), 23
- TSdata, 5, 17, 18
- TSdata.object, 18
- tsp, 25
- tsp.tfpADIdata, 25
- tsp.TSPADIdata, 25
- TSPADIdata, 1, 4–6, 9, 12, 16, 18, 21
- TSPADIdata2, 2, 3, 3